inSiDE

inSiDE • Vol. 3 No. 2 • Autumn 2005

Innovatives Supercomputing in Deutschland

Editorial

German supercomputing is going through interesting times. NIC and HLRS have installed their systems and are delivering outstanding results. LRZ has set the agenda for the next years to come and is expecting its new building and new supercomputer to come operational in 2006. Beyond the national supercomputing centers smaller centres in Dresden, Berlin, Hannover, and Karlsruhe are going through a phase of transition replacing their systems. The landscape in hardware is rapidly changing. But with kick-off of the German Grid project D-Grid the landscape is also changing with respect to new paradigms of usage and software development. inSiDE will report on first results of D-Grid in its spring issue.

The first section of this issue of inSiDE again covers application reports. The first contribution describes a prototypical way of using supercomputers coupling them to state-of-the-art visualization devices. Martin Becker presents work done by HLRS together with the Institute for Fluid Mechanics and Hydraulic Machinery on the NEC SX-8/576 system. The goal of the common project is to create a virtual workbench for the design of water turbines. Ulrich Hansmann shows results for protein folding achieved on the new Jump and Blue/Gene system of NIC that may lead the way to future breakthroughs. Gottlöber (Potsdam), Kalatyan (Potsdam), and Klypin (New Mexiko) show simulations of the Universe carried out at Jülich, Munich, and NASA. The simulation uses millions of particles to improve our theoretical understanding of the formation of different components of the observed structure of the Universe.

A novelty in high performance computing systems in Germany is presented in section two with Jülich giving an overview of its Blue/Gene system. The section furthermore deals with performance evaluation of high-end systems – a topic that has become important with the diminishing level of sustained performance that is achieved with many leading edge systems. Ebner, Brehm, and Bader from LRZ present a study on deriving characteristics of applications using hardware performance counters. With the increase of information that will be available from more performance counters in future architectures this approach will help to better understand the application side of performance. Hager (Erlangen), Bergen (LANL), Lammers (HLRS), and Wellein (Erlangen) in their contribution address the issue of bandwidth providing results for large SGI Altix systems that are impressive. Fürlinger and Gerndt from TU Munich present a system view on performance. In the frame of the DFG-funded project Periscope methods for performance analysis are investigated that are portable for very large scale systems.

As usual, this issue includes information about events in supercomputing in Germany over the last months and gives an outlook of workshops in the field. Readers are invited to participate in these workshops.

Prof. Dr. H.-G. Hegering (LRZ)
Prof. Dr. Dr. Th. Lippert (NIC)
Prof. Dr.-Ing. M. M. Resch (HLRS)

Contents

Interactive Simulation

Hydro turbines are built for a very long time. Modern water turbines are a technically mature product, their peak efficiency reaches more than 95 %. But reaching this efficiency requires a vast engineering effort, especially due to the fact that hydraulic turbines usually are individual products and thus have to be designed with respect to the local conditions, such as head and discharge. That requires a tailor-made design for the different components of the turbine.

The traditional design process focuses on experiments, measurements, and model tests. This approach leads to a time- and money-consuming development. In the last 15 years, CFD helped turbine designers in significantly shortening development times and saving money. By the use of modern computational resources, many problems can be detected in a very early development stage.

Turbine components are modelled in the computer, a computational mesh is generated and flow simulations are performed. Experiments are still very important and indispensable to validate CFD, but many of them can be replaced by simulations which play a prominent role to avoid serious mistakes in the design process.

There is a need for a tool that enables the designer to evaluate many geometry variations in a short time. To save manpower as well as money and to shorten development cycles, this tool must be based on numerical simulations. As engineers want to focus on turbine design and not on simulation details, this tool should allow working in an intuitive way.

This "virtual numerical test bed" is currently under development. It already works for some of the components. In this article, we will give an overview over the basic ideas.

Figure 1: Wicked gate of bulb turbine in Ruppoldingen power station, generator housing in the background (Aare-Tessin AG, CH).
Δ 6m, P=9.5 MW

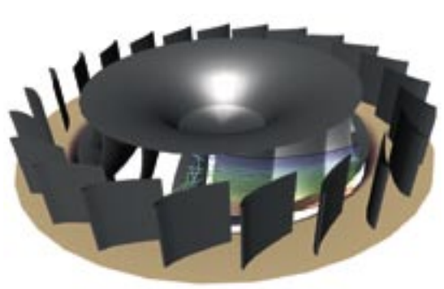


Figure 2: Francis runner and wicked gate with streamlines in virtual testbed

The Virtual Water Turbine Testbed

The virtual turbine testbed is a numerical copy of a real (physical) water turbine testbed. The water flow through an entire turbine is calculated using advanced simulation techniques. The simulation aims at predicting unwanted phenomena such as vortexes, vibrations and cavitation, which are important charges for the material and have a negative result on the efficiency.

In order to treat the "virtual turbine" in the same way as in reality, the numerics must run invisible the background. That implies the generation of computational meshes, the definition of the boundary conditions according to the operating point of the machine and the computation of the flow.

The numerics should ideally run automatically. For verification purposes however, the user should be able to check details of the simulation.

As the analysis of geometry and simulation results must be fast, detailed, and intuitive, there is a demand for a Virtual Reality Environment (VR), in which the geometry of the turbine is displayed in conjunction with the simulation results in a realistic manner. It is possible to step inside the turbine and have a closer look into every detail of the machine.

The user can have a closer look at the pressure distribution on the surfaces or can examine the flow using streamlines, particle traces or isosurfaces. That helps in detecting zones of cavitation and vortexes.

Basically, everything the user can see and measure on a real testbed should also be possible in the computer model.

Given a specific turbine geometry, the most important target is to simulate the operating behaviour for an arbitrary operating point. Thus, especially the off-design behaviour of the turbine shall be studied. Subsequently, the geometry of the machine can be changed using the experiences given by the acquired knowledge.

Due to the existence of a virtual turbine testbed, model experiments become less significant. Especially the transferability problem from model experiments to real size prototypes is solved to a

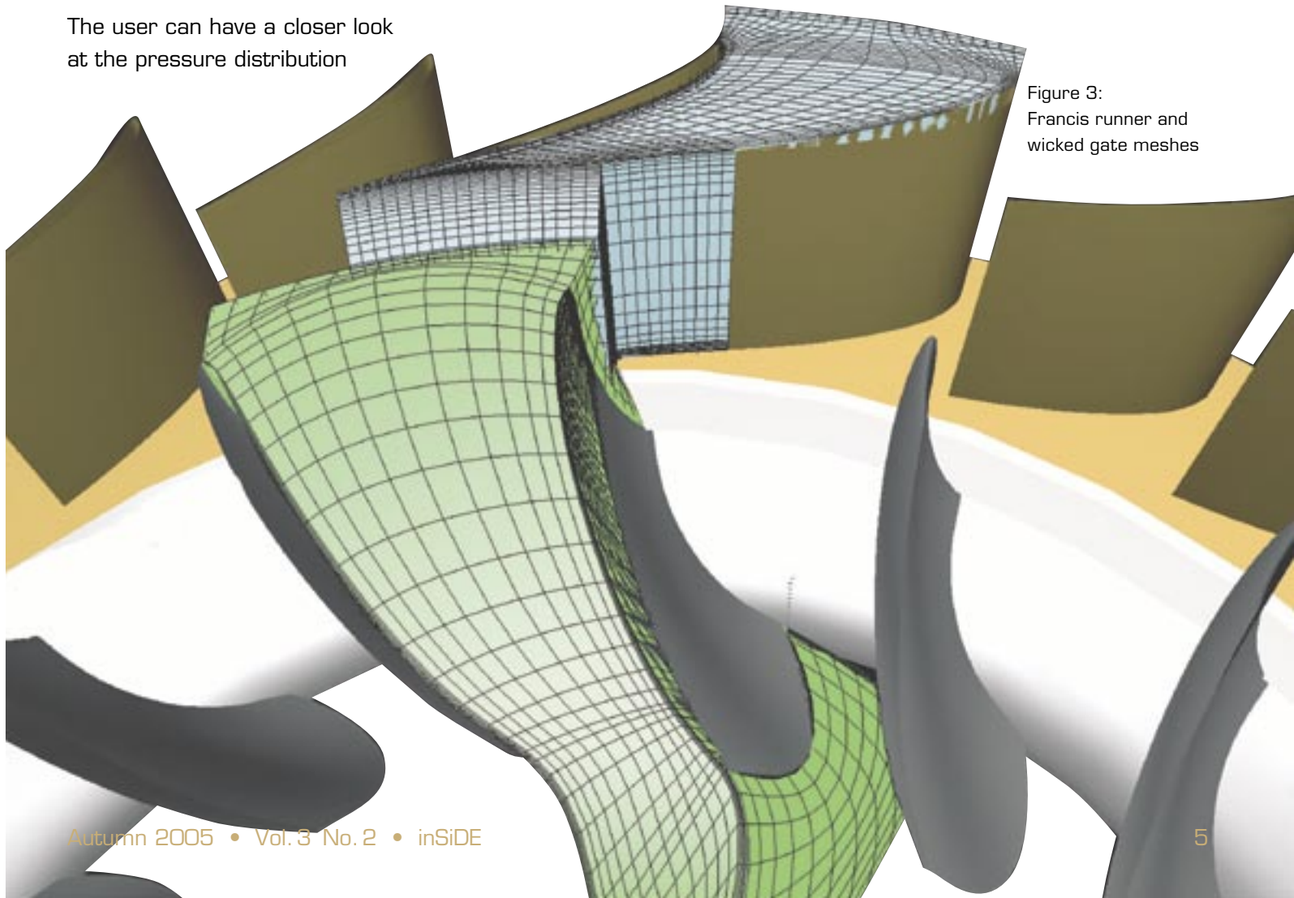


Figure 3:
Francis runner and wicked gate meshes

Applications

certain degree. The simulation of the machine can easily be done in real size without additional costs.

Design Process

In the computer-based design process, engineers in the first step define a machine geometry using complex CAD software. Afterwards, they use meshing tools that require a vast amount of expert knowledge to define the computational mesh.

Using the design modules developed at IHS (Institute of Fluid Mechanics and Hydraulic Machinery, University of Stuttgart), designing turbo machinery has become much easier.

The different modules are running under one consistent environment, COVISE, a visualization package developed at the HLRS. The whole design process is carried out from within the COVISE environment.

There are modules for all parts of the machine. These are the wicked gate, radial or axial runners and the draft tube. The turbine geometry is completely parameterized; a set of parameters describes the whole geometry. These parameters imply the shape of the runner blades and the guide vanes as well as the diameter of the runner or the moulding of the flow channel. The modules are flexible tools, some mouse clicks suffice to change these parameters and thus create a new virtual turbine that can serve as a starting setup for further geometry improvements using flow simulations.

The simulation process starts with the definition of the machine's geometry. Simultaneously, the corresponding boundary conditions that describe the operating point of the turbine are defined.

By pressing a button, an unstructured computational mesh consisting of hexahedron elements is generated.

As the modules contain automatic grid generation routines, generating a mesh does not need any user interaction. Generating a grid consisting of 100,000 elements needs about 4 seconds on modern desktop PCs.

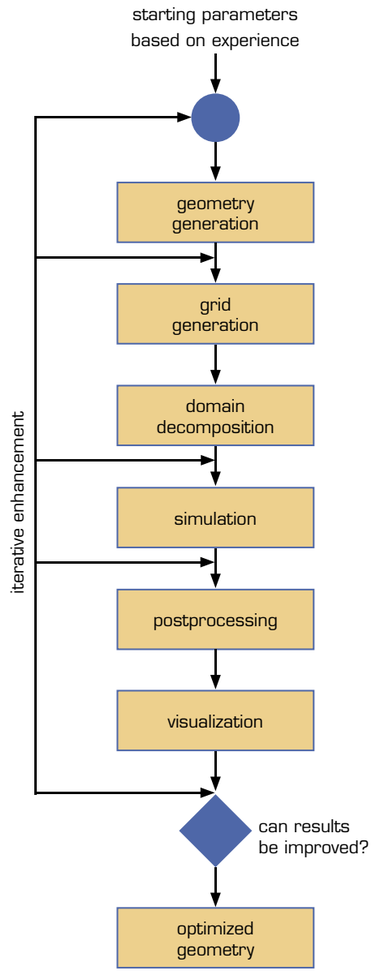


Figure 4: Process chain of iterative design

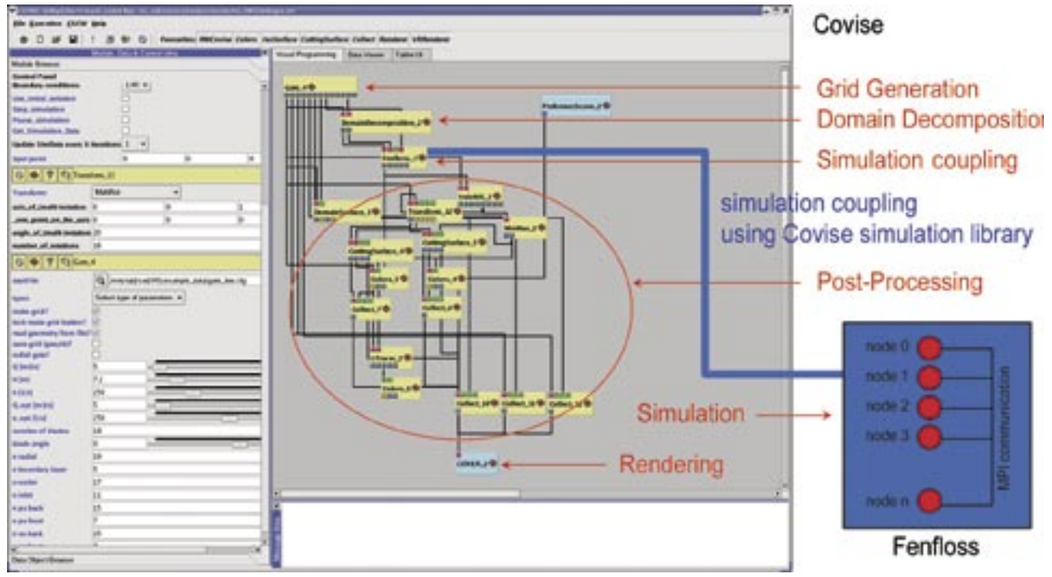


Figure 5: Process chain in COVISE as a visual program with FENFLOSS simulation coupling

In the next step, the mesh is decomposed for parallel processing.

Online Simulation

"Online simulation" means that it is possible to engage into a running simulation and that an immediate validation of the simulation results is possible. Standing inside the virtual turbine, the designer can change parameters of the machine, e.g. blade profiles or the shape of the flow channel. A new computational grid is generated immediately. Within a short time range, new stable simulation results are obtained. COVISE simulation library is used to couple FENFLOSS, the Navier-Stokes based flow solver developed at IHS, with the FENFLOSS COVISE module. As the communication uses a TCP-socket connection, the simulation part can run on any computer.

To achieve an intuitive design process, it is important to have short and almost online response times from the simulation. The engineer uses his expert knowledge to decide whether the changes led to the desired result or not and tries to optimize the turbine in an iterative process.

Of course, the intention is to obtain simulation results that are as close to reality as possible.

Since the test bed shall be used for all important components (distributor, runner, draft tube), the use of massively parallel code on supercomputers is essential.

COVISE

We use the COVISE visualization system to integrate the whole process from grid generation, simulation to analysis. Each of the modules in this workflow can reside on a different computer. This allows distributing the work load among different machines. Typically, the pre- and post-processing modules run on a visualization server while the simulation runs on a remote supercomputer. The display modules either run on a visualization cluster

which drives a CAVE or tiled wall or on the workstation of a user.

COVISE is a modular visualization system developed at the HLRS. The software uses a data-flow execution model, i.e. the data objects in COVISE flow through a network of modules. The modules all run as separate processes and thus can be distributed among multiple computers.

It is possible to couple COVISE environments for collaborative working.

COVISE can be used not only for off-line post-processing and visualization; it can also be used as a general distributed and collaborative integration platform.

This allows integrating grid generation, simulation and post-processing modules in a seamless way to create interactive engineering applications.

The simulation process chain consists of three COVISE modules which have been developed to integrate all the aforementioned processing steps into one environment.

At first we need a module that defines the geometry and generates the grid and boundary conditions. This can be the Gate module, one of the runner modules or the Draft Tube module. Next, the Domain Decomposition module, which decomposes the grid into multiple domains for parallel simulation, and the

Figure 6: Kiebingen power station

Figure 7: Kiebingen power station in VR

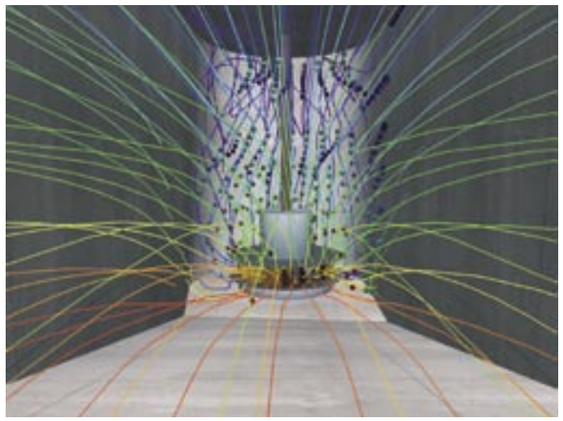


Figure 8: Inside inlet building of Kiebingen power station in VR, wicked gate in the background

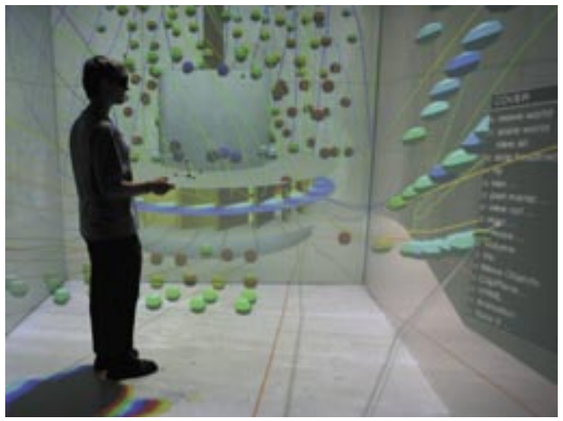


Figure 9: Online simulation of water flow in Kiebingen wicked gate in the CAVE

FENFLOSS module, which couples the simulation code to COVISE. The entire COVISE dataflow network is shown in Figure 5. The simulation itself is a separate process that is coupled with the FENFLOSS COVISE module using a socket connection. It sends new data to COVISE after each global iteration. All the other modules in Figure 5 are used for data analysis and visualization, e.g. Tracer and Cutting Surface modules.

Application Example: Kiebingen Power Station

Modern CFD was used to increase the efficiency of Kiebingen power station, a small plant situated at the river Neckar in Southern Germany. The four machines, that are aged almost a century, needed some redesign. The power output per machine could be increased by 30 %, simply by substituting runner and wicked gate. All other parts were retained.

Figure 6-Figure 9 show Kiebingen power station in real world and in VR, together with simulation data.

Tangible Interfaces

As a further enhancement of the described simulation techniques, we have developed an airflow simulation in urban planning using a tangible interface as an easy and effective solution for interaction.

The user can control the simulation and visualization through manipulation of physical objects. These movable objects are equipped with black and white markers which identify the object. A high resolution IEEE1394 Camera is positioned above a physical model of the simulation domain. The camera picture is captured and analyzed by a modified version of ARToolKit (developed by Hit Labs, University of Washington). It

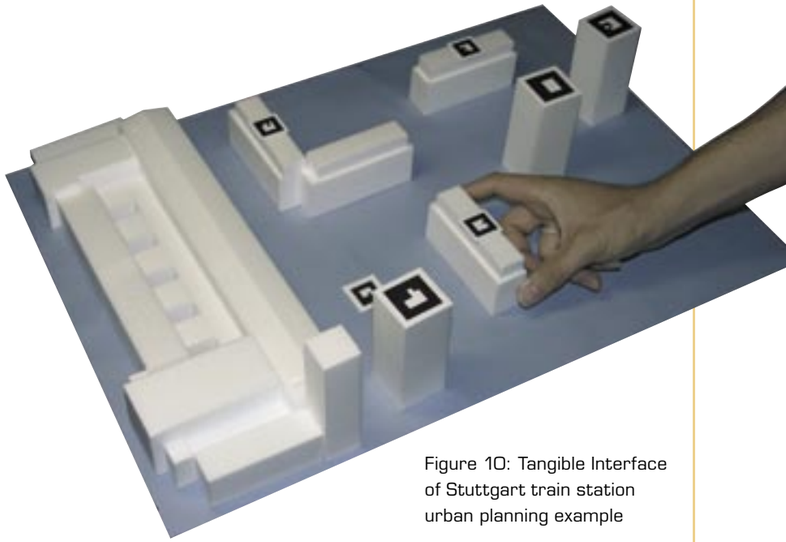


Figure 10: Tangible Interface of Stuttgart train station urban planning example

returns the position and orientation in six degrees of freedom of all markers which are completely visible.

Tangible interfaces can be used as controls for digital parameters, data sets, computing resources, and other digital content. There are many advantages of this approach. First of all, the parameters that can be changed using the tangible objects are represented in a very clear way. The interaction is simple due to the fact that it is so close to reality. In addition to that, tangible interfaces can be used on desktop applications as well as in immersive environments.

Users can explore the solution collaboratively by sharing multiple screens or immersive environments distributed in a room, building or anywhere around the globe. Results of the simulation are available in real-time.

A VRML model of the objects which are simulated serves as a visual reference in the virtual environment and it defines the tangible interface. A new VRML node, ARSensor, acts as interface between ARToolKit and the VRML model.

It takes the position and orientation of one of the markers from ARToolKit and transforms it into the VRML coordinate system. This transformation can then be routed to the transform node of the tracked object which thus follows the movement of the marker and the physical object respectively.

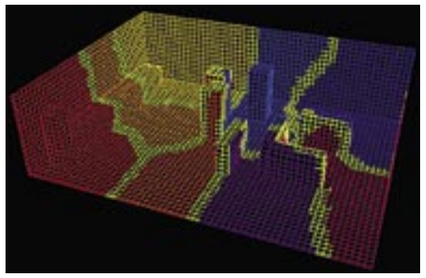


Figure 11: Grid around urban planning buildings. Each partition is dyed differently

An automatic grid generator receives the position of the objects and generates a hexahedronal computational mesh around the objects which is then decomposed for parallel processing. Figure 11 shows the decomposed grid with each partition coloured different.

By using tangible interfaces, a much more natural mode of operation is possible than with a traditional user interface where it is usually necessary to type in coordinates to move objects. The natural perception of the model in comparison to a simple monitor image is another important advantage.

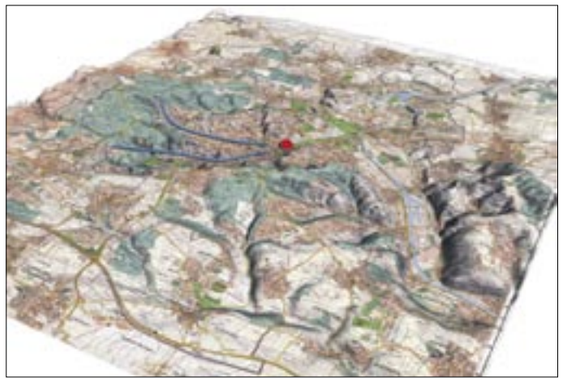


Figure 12: Topography of Stuttgart (z-scaled)

Urban Planning Example

In an interactive airflow simulation of downtown Stuttgart, we simulate the impact of architecture on the fresh air supply of the city.

Stuttgart is located in a basin. In Figure 12 the topography of the area is shown. The red sphere represents the downtown district around Stuttgart main station and the castles.

The rail-tracks will be removed and put underground within the next years. This will reveal a huge site for new buildings right at the heart of Stuttgart. As there are hills almost all around the city, the area is an important corridor for the city's fresh air supply (see blue arrows) and thus is of significant importance for the Stuttgart locale climate.

The challenging task is to find a construction plan which bears that in mind.

The dataset covers an area of 250 x 300 m of the area behind the Stuttgart railway station. The boundary condition is a wind profile with low speed on the ground and higher speed towards the top of the grid. The wind direction can be changed interactively. Six buildings can be placed in that area by either changing parameters of the grid generation module or by moving the physical buildings in the model.

We have built a model of the construction site (Figure 10). The airflow around these buildings is simulated using our described approach. By moving the modelled buildings around, many different layouts can be tested in a very short time while at the same time the user gets a vivid impression of the architecture in the virtual environment. Cutting planes and particle traces can be positioned by moving a marker in the model or in the virtual environment with the 3D mouse.

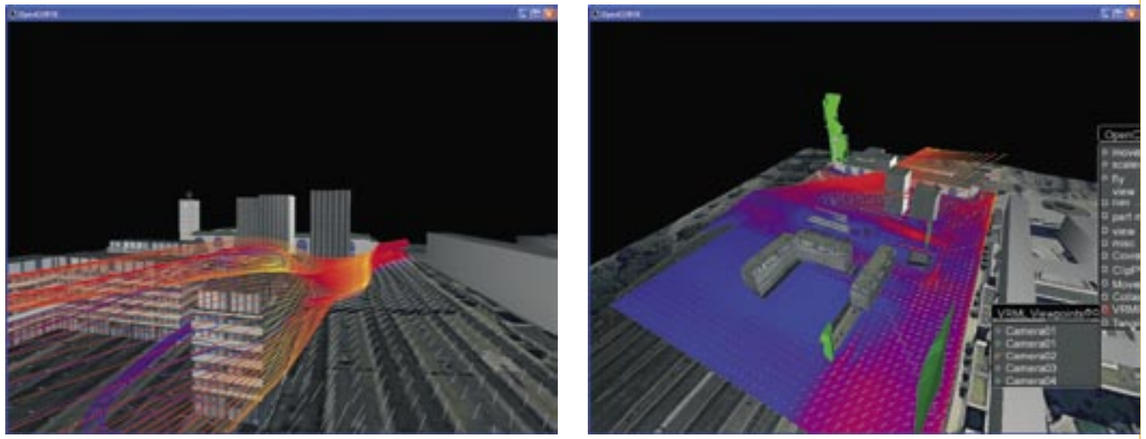
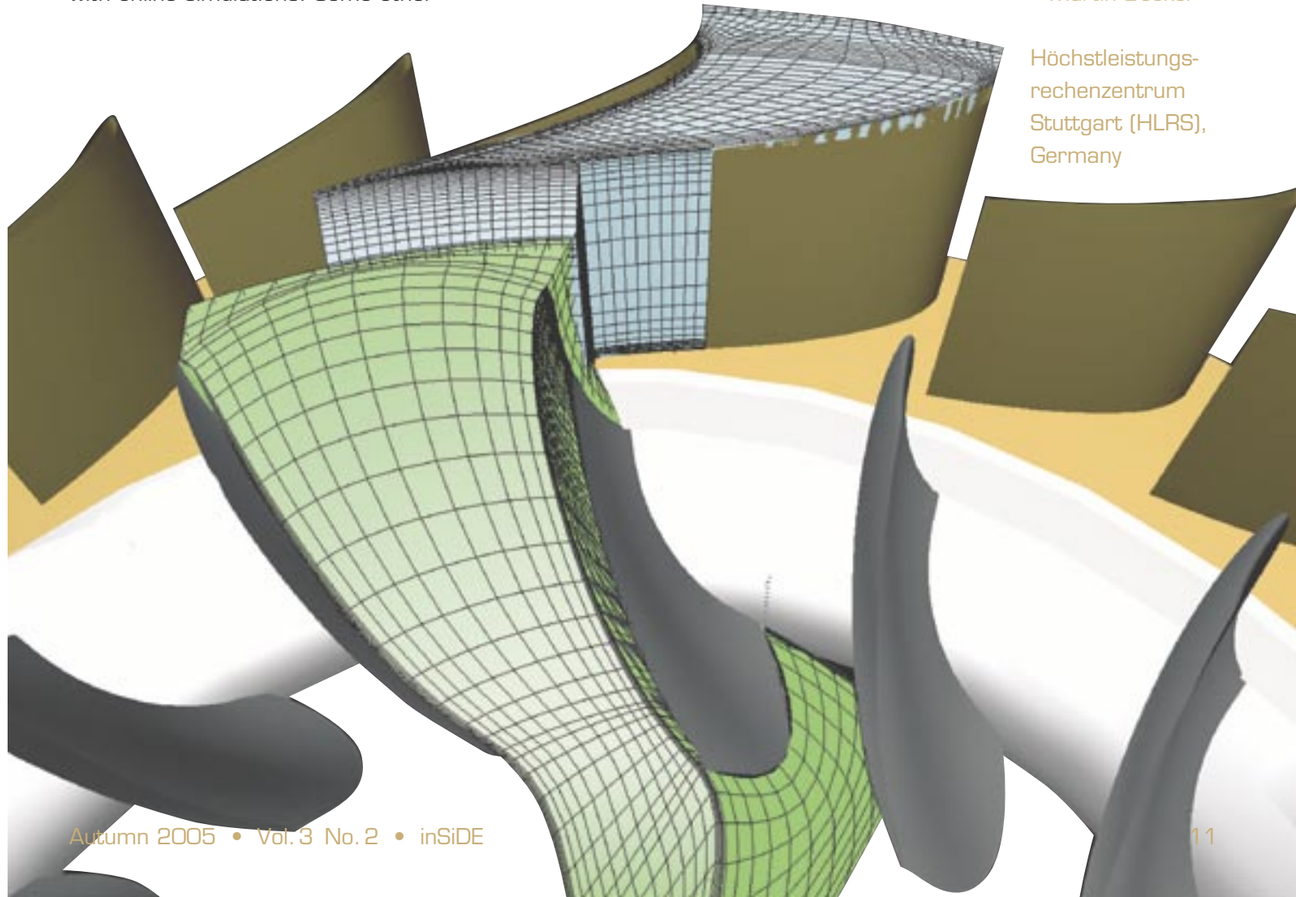


Figure 13: Screenshots of the VR-Display, cutting-surfaces, particle-traces and isosurfaces

This is just one example for using tangible interfaces in conjunction with online simulations. Some other possible application scenarios for this simple grid generator are the layout of clean rooms, where a laminar airflow is required or the allocation of supercomputer racks in a computing room where it is important to design and dimension the air-condtion.

• Martin Becker
Höchstleistungsrechenzentrum Stuttgart (HLRS), Germany



Protein Folding in Silico

With the recent successful completion of the Human Genome Project and related attempts to determine whole genomes it has become obvious that the obtained wealth of data needs to be matched by information on the function and interaction of the huge number of encoded proteins. These polymers are the workhorses in a cell and are responsible for transporting molecules, catalyzing biochemical reactions, or fighting infections.

Proteins are only functional if they assume specific shapes. Despite decades of research it is still an open question how these structures emerge from a protein's chemical composition (the sequence of amino acids as specified in the genome). An answer to this question could lead to a deeper understanding of various diseases that are caused by the miss-folding of proteins, and enable the design of novel drugs with customized properties.

Given a sufficiently accurate description of the forces between the atoms in a protein, and between a protein and the surrounding environment, it is theoretically possible to simulate the folding of a protein. However, the complex type of the interactions containing both repulsive and attractive terms leads to very rough energy landscapes. Hence, sampling of low-energy conformations becomes a hard computational task.

Only with the use of massively parallel computers (such as JUMP and the new BlueGene in Jülich) and the development of advanced simulation techniques are we approaching a point where atomistic simulations of stable domains in proteins (usually of order 50-200 residues) become feasible. Sampling in generalized ensembles, parallel tempering and energy landscape paving are some of the novel algorithms that allow exploring low-energy configurations without the simulations getting trapped in a local minimum.

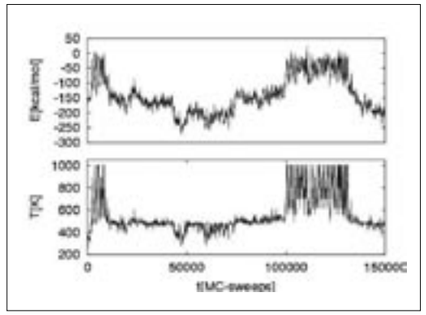


Figure 1: Time series of energy and temperature of one replica in a parallel tempering simulation of the 36-residue protein HP-36

Especially interesting for parallel computing is parallel tempering where one distributes on the nodes of a multiprocessor machine N copies of the molecule, each at a different temperature. In addition to standard Monte Carlo or molecular dynamics moves on each copy, parallel tempering allows with certain probability the exchange of conformations between two copies i and i+1. I show as an example in Figure 1 the time series of temperature and energy of one arbitrarily chosen replica as obtained in a parallel tempering simulation of the 36-residue protein HP-36 (the figure is taken from [1]). Note how the resulting random walk in temperature leads to one in energy that enables escapes out of local minima. In this way sampling of low-energy structures will be enhanced. A simple implementation of this and other modern protein simulation techniques can be found in the free program package SMMP (Simple Molecular

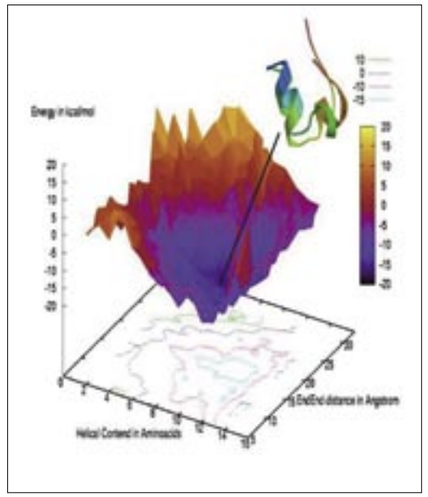


Figure 2: Energy landscape of the trp-cage protein

Mechanics for Proteins) which is available from the web page [2].

Current applications focus on probing the mechanism of folding in small proteins and the conditions under which proteins mis-fold and aggregate. A now widely accepted assumption is that a protein evolves into its biologically active structure through a diffusive process along a funnel-shaped energy landscape. Figure 2 shows as an example a two-dimensional projection of the folding funnel of the 20-residue trp-cage protein as determined in a computer simulation. Configurations found at the bottom of the funnel resemble closely the experimentally determined structure (the figure is taken from [3]).

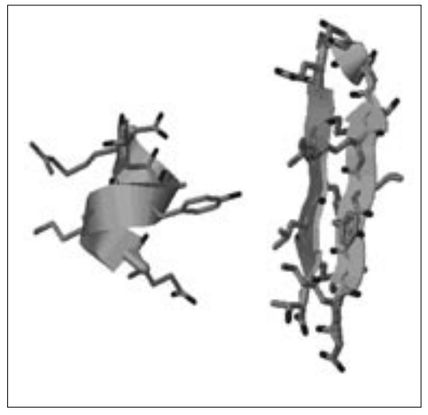


Figure 3a and 3b: Low-energy configurations of the chameleon peptide EKALYRT

Particularly interesting and important are situations where proteins fold incorrectly. Formation of a β-strand instead of an α-helix in part of a protein can lead to aggregation and formation of fibrils that are often related to the outbreak of neurological diseases. A possible mechanism for the growth of the toxic fibrils may be that the incorrectly folded protein induces mis-folding in close-by molecules. For instance, the peptide EKAYLRT likes to form a β-strand when in the vicinity of an other β-strand (Figure 3b), while further away (or isolated) it tends to form an α-helix (Figure 3a). The figure is taken from [4].

One challenge in the next years will be to extend these lines of research to larger and medically relevant proteins. Other research will focus on the interaction of proteins with different biological molecules (flexible docking) in order to understand how biomolecules interact and regulate each other in a cell. Applications of the current research may also include the use of proteins for assembling nanostructures.

References

[1] C.Y. Lin, C.-K. Hu, U.H.E. Hansmann
Proteins 52 (2003) 436

[2] www.phy.mtu.edu/biophys/smmp.htm

[3] A. Schug, W. Wenzel, U.H.E. Hansmann
J. Chem. Phys., 122 (2005) 194711

[4] Y. Peng, U.H.E. Hansmann
Phys. Rev. E, 68 (2003) 041911

• Ulrich H.E. Hansmann

John von Neumann
Institute for Computing
Research Group
Computational Biology
and Biophysics
Research Centre Jülich

The Universe on Small Scales

During the last 10 years new extensive observations of the Universe were made using both ground-based telescopes and space instruments. These measurements have provided new insights into the structure of the Universe on various scales. A wide range of the electromagnetic spectrum emitted by cosmic objects has been studied. The wavelengths extend from very long radio wavelengths to energetic gamma rays. This observational progress has been accompanied by considerable effort in our theoretical understanding of the formation of different components of the observed structure of the Universe: galaxies and their satellites, clusters of galaxies, and superclusters. A substantial part of this theoretical progress is due to the increasing possibilities of using ever improving numerical models, which mimic the structure formation on different scales using the new generation of massively parallel supercomputers. Looking back in recent history (20 years or so), it is interesting to note that the resolution of the numerical simulations roughly followed Moore's Law. In the eighties the best simulations handled about 32^3 particles, whereas now we can reach 1024^3 particles, a factor of 215 increase during 20 years. Moore's Law predicts a factor of 213. The remaining factor of 22 in speedup is due to more sophisticated numerical algorithms. The dramatic increase in the accuracy and quality of modeling is astonishing.

The effort of observers and theorists brought about the so called concordance or standard cosmological model. This model is based on the idea that some kind of dark energy contributes about 70 % of the total energy density of the spatially flat Universe. The simplest form of the dark energy is the cosmological constant, which was introduced in 1917 by Albert Einstein in his paper about treating the cosmological solutions of the field equations of general relativity. The remaining 30 % of energy density consists of matter. About 85 % of this matter is made of unknown dark matter particles, the remaining 15 % is the contribution of "normal" baryonic particles well known to particle physicists. This means that the nature of more than 95 % of the matter in the Universe is not yet understood.

According to the standard cosmological model, the main process responsible for the formation of observed structures is gravitational instability. The initial seeds, which eventually became galaxies and superclusters and all other structures, resulted from the quantum fluctuations generated during the early inflationary phase: ~ 10^{-35} sec or so from the beginning of the Big Bang. The power spectrum of these primordial fluctuations has been confirmed by measuring the temperature fluctuations of the cosmic microwave background radiation. These temperature fluctuations tell us the magnitude of the small fluctuations in the Universe about 300,000 years after the Big Bang.

One of the key features of the standard model is its simplicity. The expansion rate and the clustering properties are described by only few parameters which can be measured with quite high accuracy. Moreover, there are no preferred length scales in the model. Thus, structure formation is predicted to be essentially scale invariant: in a statistical sense the structures on scales of galaxy clusters are repeated on scales of galaxies. Typically, small objects merge together and form more and more massive objects. However, the small objects do not disappear within those larger objects but rather form a complex hierarchy of substructures. This hierarchical scenario predicts that our Milky Way Galaxy is expected to have as many satellites (many hundreds) as a cluster of galaxies has galaxies. However, the Milky Way has only a dozen satellite galaxies: a far cry from what is predicted.

Altogether, we arrive at a picture in which dark matter particles form the backbone structure for all objects in the Universe from clusters of galaxies to dwarf satellite galaxies. Normal matter (baryons) falls into the potential wells formed by the dark matter particles and forms the luminous objects. The details of this formation process must be followed using hydrodynamical simulations. However, many features can already be studied by semi-analytical methods which are based on the evolution of the dark matter halos as measured in the dark matter simulations.

The nonlinear evolution of cosmological fluctuations can be studied only numerically. The requirements for modern cosmological simulations are extreme: a very large dynamical range for force resolution and many millions of particles are needed. These requirements are just a reflection of the vast range of masses and spatial scales in real astronomical objects. For example, from dwarf galaxies to galaxy clusters the mass spans about seven orders of magnitude. The range of scales is also enormous: from the inner structure of galaxies (sub-kiloparsec scales) to cosmological distances of many megaparsecs (1 pc = 3.26 light years).

We have developed a highly efficient parallel Adaptive Refinement Tree (ART) code, which tracks the evolution of small-amplitude perturbations from the early Universe until the present time. Our code provides a dynamical range of up to 500,000 and handles millions of particles of different masses. The code also includes hydrodynamics. At present, for very large simulations we use a hybrid MPI-OpenMP mode of parallelization. In this mode the code can handle up to 1 billion particles.

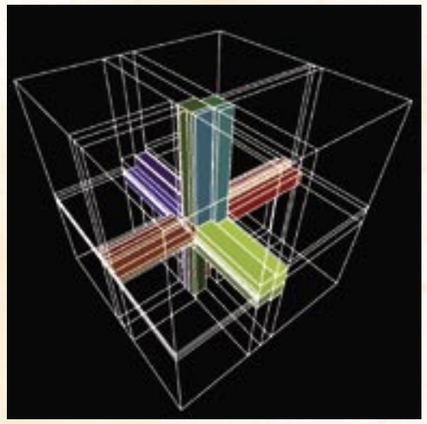


Figure 1: Decomposition of the simulation box into 64 domains with each domain handled by one MPI task. The code decreases the volume of domains in regions with many particles and high force resolution (the center of the box) to achieve load balance. The colored 24 smallest domains cover the central region

The first step of running cosmological simulations is to set up the initial conditions: amplitudes and phases of small perturbations at very high redshifts. Having in mind that the largest structures in the Universe – superclusters and voids – have sizes of 10-50 Mpc, the simulated volume should be significantly larger than a supercluster or a void. However, we may be interested

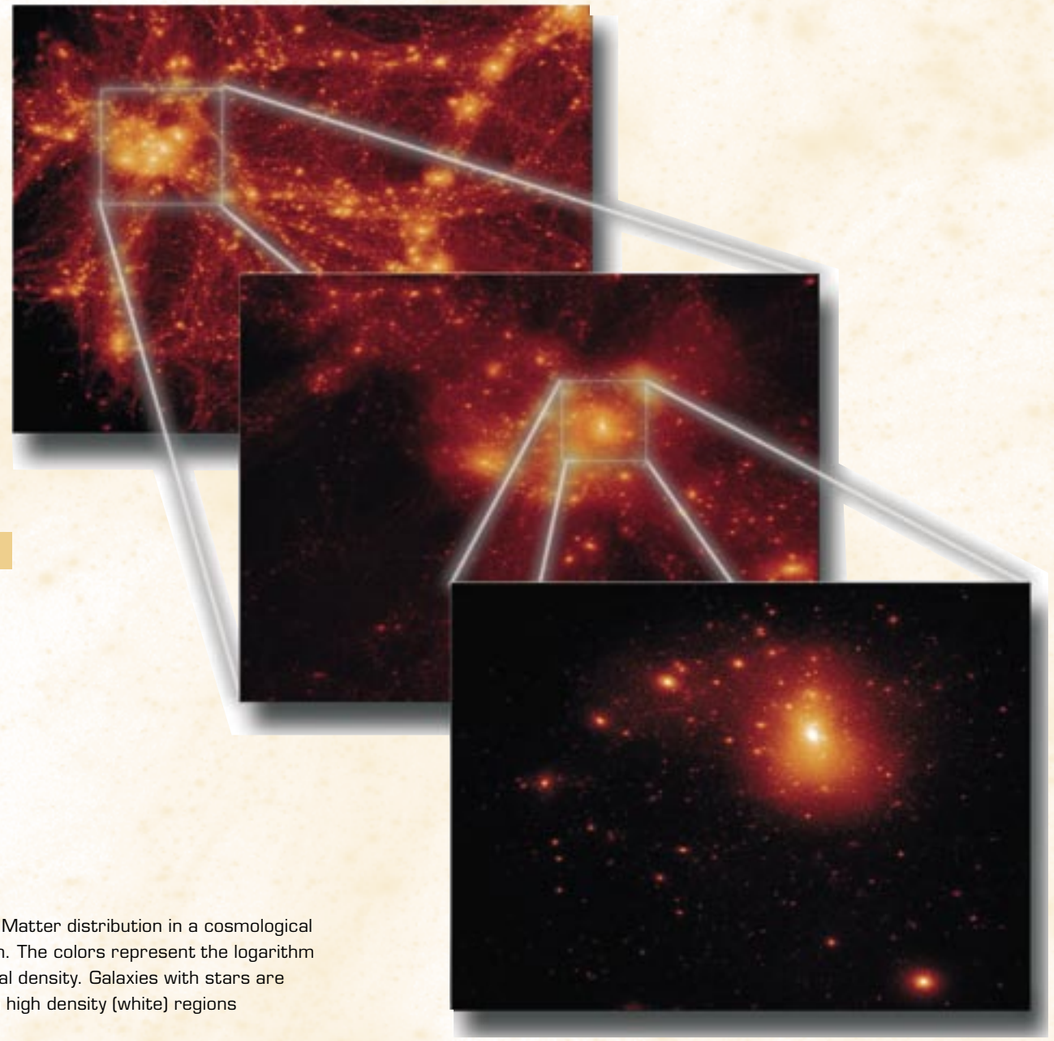


Figure 2: Matter distribution in a cosmological simulation. The colors represent the logarithm of the local density. Galaxies with stars are formed in high density (white) regions

in the structure of a much smaller object such as our Milky Way Galaxy or its satellites. In N-body simulations each mass element is represented by a point-like particle. The mass resolution is limited by the total number of particles computers can handle at a given time. Thus, increasing the representative cosmological volume decreases the mass resolution. To overcome this problem we developed a mass refinement technique: To construct suitable initial conditions, we first create an unconstrained random realization at highest possible resolution, i.e. 2048^3 (~ 8.6 billion) particles. The initial displacements and velocities of N particles are calculated using all waves ranging from the fundamental mode $k=2\pi/Lbox$ to the Nyquist frequency $kNy=2\pi/Lbox \times N1/3/2$. We then merge (lump together) particles, assigning to merged particles a velocity and a displacement equal to the average values of the original small-mass particles. Using the smaller number of more massive merged particles we first run low-resolution simulations until the present epoch.

In this simulation we select the regions of interest. Then we repeat the process by starting from refined initial conditions. This time we preserve the original, very small-mass particles but only inside the region, which later collapses to produce our object of interest. Outside of this region we progressively merge more and more of the small particles creating shells of larger and larger particles. This procedure ensures that our object evolves in the proper cosmological environment and with the right gravitational tidal fields.

We have used the technique described above to simulate the evolution of a region similar to our cosmological neighborhood. In this simulation we use more than 150 million particles within the high resolution region. The mass resolution is 5 x 106 M0 and hence a Milky Way Galaxy is represented by 200,000 particles. The force resolution reaches 300 pc. The total CPU time used for this simulation was about 300,000 CPU hours. The initial conditions (they need 105 Gb of shared memory) have been calculated on the SP4 at NIC Jülich. A substantial part of the simulation has been done using 512 CPUs of the Hitachi SR 8000 at Leibniz Rechenzentrum Munich, the rest at NASA's SGI Altix 3000.

Figure 2 shows a region of 35 Mpc size. It is a small part (3 %) of a larger simulation with a total volume of $(115\ Mpc)^3$. One can clearly see the filamentary structure. The first blow-up shows a region of 10 Mpc and the second one of 2 Mpc. The object in the second blow-up has a mass comparable to our Milky Way.

Even though high resolution simulations already allow the comparison of many theoretical predictions with observed properties of galaxies in various environments they are still far from providing us with a complete understanding of the processes of galaxy formation. In the near future observations will deliver detailed information about the inner structure of galaxies and the formation and evolution of low mass objects, in particular about such objects at high redshift, which are the progenitors of the present-day galaxies. These improved observational data sets need to be accompanied by an equally sophisticated theoretical and numerical modeling of structure formation. To this end one also needs to follow the evolution of the gas and the formation of stars in cold clouds; to know their impact on the interstellar medium and on the formation of the observed stellar disks and bulges. These models become more and more complex and span an even wider dynamical range. To cope with such tasks the largest available supercomputers will be required by astrophysicists working in numerical cosmology.

• Stefan Gottlöber
• Arman Khalatyan

Astrophysikalisches Institut Potsdam

• Anatoly Klypin

New Mexico State University, Las Cruces, USA

IBM Blue Gene/L in Jülich: A First Step to Petascale Computing

In early summer 2005, JUBL (JUlich Blue Gene/L), an IBM Blue Gene/L system, was installed at the Research Centre Jülich as the first one of its kind in Germany. At that time, a larger Blue Gene/L system was No. 1 on the Top 500 list. The Blue Gene/L has an innovative architecture and is used by well-known institutions like Lawrence Livermore National Laboratory, Argonne National Laboratory, San Diego Supercomputing Center, and EPF Lausanne for selected applications in high performance computing.

The commitment to this system in Jülich was aimed at the analysis of the architecture of a future leadership-class system which can be extended to very high processor numbers. In comparison to a general purpose supercomputer this system is very competitively priced with noticeably reduced cooling and foot print requirements while maintaining a comparable peak performance. On the other hand only very high scaling applications can exploit the full capacity of the current Blue Gene system meeting at the same time some memory and kernel restrictions. Jülich wanted to give potential users the earliest possible opportunity to determine whether their applications are suited to run efficiently on JUBL. So appropriate applications were selected from a variety of science areas and are now being tested with respect to portability and efficiency.

JUBL consists of one Blue Gene/L rack with 1024 compute nodes, each equipped with two IBM PowerPC 440 processors including additional floating-point units (700 MHz, 2.8 GFlops peak performance) and 512 MB memory. The peak performance of JUBL thus is 5.6 TFlops and the LINPACK performance was measured at 4.7 TFlops. In the Top 500 list of summer 2005, this system is ranked #60.

According to their specific requirements, the applications can use either the coprocessor mode, where one CPU in a node is used exclusively for MPI communication, or the virtual shared node mode, where both processors run independent MPI tasks. In addition to the compute nodes the system has 64 I/O nodes with an external 1 GB/s Ethernet adapter each. There are five different connection networks. For applications, the 3D-torus network and the tree-topology network are important. The latter is dedicated to collective MPI operations. The torus has a maximum latency of 6.4 microseconds and an aggregate bandwidth of 2.1 GByte/s per node. The compute nodes run a reduced Linux kernel without any time-slicing and paging capability whereas the I/O nodes run a complete Linux kernel. The system complex is completed with two larger nodes comprises also one Front-End node for user access and a Service node for system operations. Both run with a standard SuSE-Linux operating system.

The I/O nodes of JUBL have access to an external 2-TByte file system for user data and applications. It is planned that in the near future JUBL can access the HOME file systems of Jülich's IBM p690 supercomputer JUMP (see inSiDE Vol. 2 No. 1) as GPFS client to enable users to access their data from both supercomputers. An additional advantage of this solution is that users get access to the STK data robot systems. The aggregated peak performance of both supercomputers in Jülich is 14.5 TFlops.

Although the system was not available for applications before July, there were about ten serious enquiries in June and July from projects already successfully peer-reviewed and running on JUMP, aiming at capability computing. For first tests, four of them were selected, coming from the research fields lattice quantum chromodynamics (QCD), computational chemistry (Vienna Ab initio Simulation Package, VASP), materials science (Dynamical Mean Field Theory, DMFT), and laser-plasma interaction. All groups have successfully ported and run their codes within a short time, facing only minor problems. It was also possible to establish an early production environment for them, which allows a smooth operation for a handful of users. The applications used JUBL nearly to its full capacity, requesting between 128 and 512 nodes for production runs. Depending on the problem size, the runs were are either performed in coprocessor mode or in virtual node mode.

Comparing preliminary BG/L performance results from these applications with corresponding results obtained on the JUMP system, first estimates can be presented: Considering that a single Blue Gene/L processor has 41 % of the peak performance of a single Power 4+ processor of the JUMP system, the structure optimizations carried out on BG/L with the VASP code reached between 22 % and 43 % of the corresponding JUMP code depending on the chosen structures, the DMFT and the laser-plasma calculations were around 33 %. The best performance – 50 % of a Power4+ processor – was measured with the QCD code.

Looking at the scaling plots of this application the measurements become even more impressive. The major subroutine of the QCD code (Wilson operator) reached a scaling factor of nearly 4 upgrading going from 32 to 128 processors and a factor of nearly 16 upgrading going from 32 to 512 processors. The members of the QCD project at Jülich are very satisfied with these results showing a nearly 100 % scaling with these processor numbers. They expect further performance improvements from already planned code reorganizations and from announced system kernel modifications.

For more information please see
www.fz-juelich.de/zam/ibm-bgl

Figure 1: IBM Blue Gene/L

• Dr. Norbert Attig
• Klaus Wolkersdorfer

Central Institute for Applied Mathematics (ZAM),
Research Centre Jülich

Systems

Deriving Characteristics of Applications by using Hardware Performance Counters

Although programming models and languages appear to be converging, the computational workloads and communication patterns for scientific applications vary dramatically, depending in part on the nature of the problem the applications are solving.

Typical job accounting does not provide sufficient information about the characteristics of applications running on HPC systems. It is also impractical to use trace-based tools to monitor the behaviour of all applications on a system. Instrumented versions of the MPI library can be used to provide a detailed summary of the hardware performance counters and of the MPI calls, but this produces no immediate information during the run time of a job.

LRZ uses a more general approach to monitor all applications on its HPC systems. Samples of the most important hardware counters are taken from all nodes in 5 minute intervals, and are stored and subsequently processed in a database. On the Hitachi SR 8000, the hardware counters and information from processor and communication network, listed in Table 1, are used for the analysis.

Users and support staff are automatically informed when an application falls below predefined values or combinations of parameters which hints at severe performance problems in the application itself. Also users can query the database for the performance of any job run in the present or past, and detect how the performance changed due to their modifications, or to get a qualitative global perception of the application behaviour. Furthermore, for the computing center itself such measurements contain important information for future procurements of new systems.

Here, we now want to present an analysis of data collected for all jobs

User and System CPU Time
Memory Usage
Number of Instructions (Memory/Floating Point/Integer)
Number of Load/Store Instructions
Number of Data/Instruction-TLB Misses
Number of Data/Instruction-Cache Misses
Number of Floating Instructions
Number of System-calls
Number of Packets and Bytes sent or received on a Node
Barriers (Time and Number)
Data on Pages, I/O etc.

Table 1: HW counters on the SR 8000

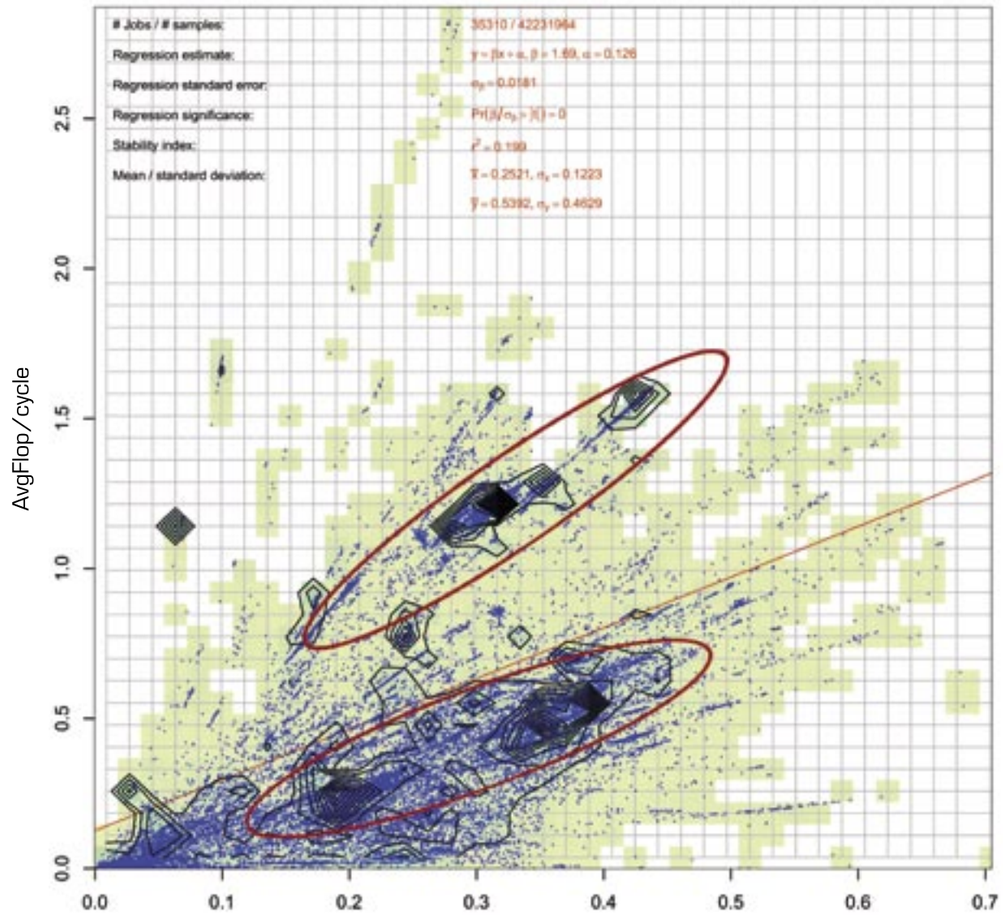


Figure 1: LdSt vs. Flops for all application areas

on the SR 8000 system over a period of four years. About 35,000 jobs have been processed, and more than 42 million samples have been taken during this period.

One of the most important metrics for the performance of applications is the computational intensity, i.e. the ratio of Load/Store operations (LdSt) and Floating Point operations (Flops).

In Figure 1 we depict the number of LdSt per cycle vs. the number of floating point operations per cycle on a single CPU basis. Every blue dot represents a measurement. Due to the limited resolution many points appear to conincide. Therefore we have also plotted contours of the frequency of occurrence. The general regression analysis yields a slope of 1.7 Flops per LdSt operation (thin red line), but there are two separated islands (indicated by red ellipses) where most of the points cluster. For the upper region we get roughly 4 Flop/LdSt while in the lower region we have about 1.6 Flop/LdSt. Looking deeper into the user codes, we can assign the upper region to optimal library routines, e.g. for BLAS and FFT operations.

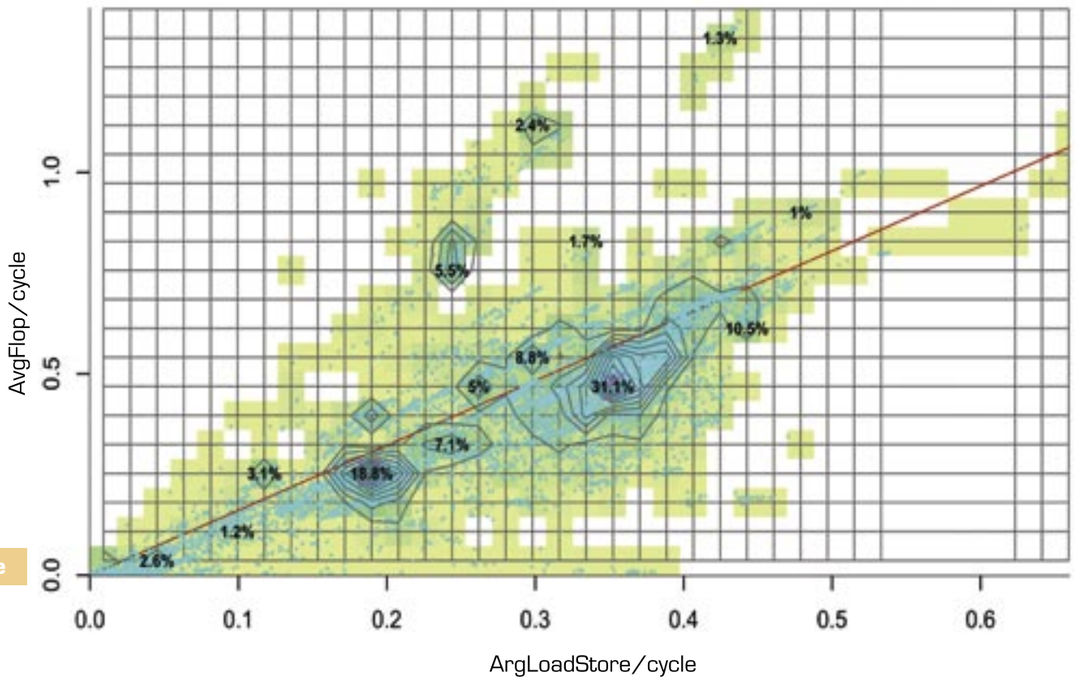


Figure 2: LdSt vs. Flops for CFD jobs only. Slope is 1.6 Flop per Load/Store

If we limit our statistical samples to fluid dynamics applications (Figure 2), we get a rather different picture, since we now see only one region with a slope of 1.6.

Typical ratios for other application fields are summarized in Table 1. It is clear that the performance of most jobs is limited by the available memory bandwidth; only for the optimized library usage the required bandwidth roughly matched the available one on the machine.

Application Field	Flops per Load/Store	Required Bandwidth Byte/Flop
All Jobs	1.7	6.6
Optimized Libs	4.0	2.8
Geophysics	1.0	11.2
CFD	1.6	7.0
Chemistry	2.0	5.6
Solid State Physics	2.4	4.7

Table 1: Computational intensity for various application areas (roughly 70 % of LoadStores are actually Quad-LoadStores). The theoretical bandwidth of the SR 8000 is 2.6 Byte/Flop

In Figure 3 we have illustrated the cumulative ratios of the sustained Flop/s vs. the sustained MPI traffic (Bytes/s) across the internal network interconnect, a crossbar switch. Having to decide what is an appropriate ratio for a well-balanced system, we came to the conclusion that 0.1 Byte/s per sustained Flop/s is sufficient for most applications (look for the arrow in Figure 3) since more than 85 % of the time the applications' requirements are below that value. Furthermore, requiring higher bandwidths would incur costs quite out of proportion to further gains on code performance. As the memory bandwidth is the limiting factor for most applications (see above), we compared this with the theoretical bandwidth of the SR 8000 (32 GByte/s vs. 12 GFlop/s per node). We conclude that an appropriate ratio between (unidirectional) network interconnect and memory bandwidth is of the order of 1:30.

Looking at the ratio of LdSt to DCache Misses (Figure 4) we can conclude that the software pipelining as well as the preload and prefetch mechanisms of the SR 8000 are very efficient in hiding latency. The majority of codes show only ratios of the order of a few cache misses per 1000 LdSt. Here we note, that there is only a small 128 KByte L1 cache available on the Hitachi processor but enhanced prefetching mechanisms are implemented, which are obviously able to prefetch the data to the L1 cache ahead issuing of the corresponding Load instruction.

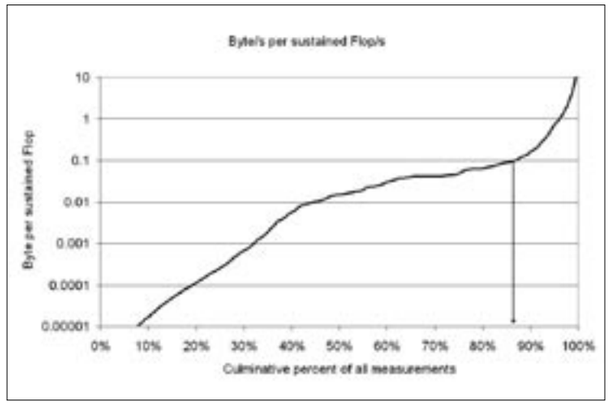


Figure 3: Cumulative distribution of MPI Bytes sent through the communication network per sustained Flop

The new Itanium-based systems, which will be the successor machines for the SR 8000, provide far more performance counters. Similar analyses will be performed on this follow-on architecture in the future.

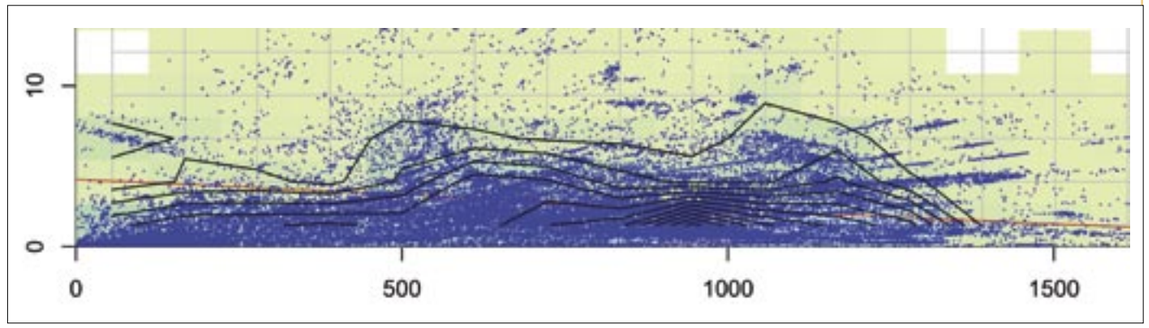


Figure 4: LdSt vs. Cache Misses. Contour lines show the relative frequency

• Ralf Ebner
• Matthias Brehm
• Reinhold Bader

Leibniz Computing Centre, Munich

Taming the Bandwidth Behemoth First Experiences on a Large SGI Altix System

Abstract

The upcoming HLRB 2 installation at LRZ Munich with more than 5000 Itanium 2 CPUs will force many developers to boldly go where no programmer has gone before. Scaling scientific applications to thousands of processors is a challenging task even on a machine that provides seemingly endless bandwidth resources. Consequently, the number of codes that can put such a system to full use is limited today. Thanks to SGI, the Erlangen HPC group were able to test-run two such applications on a 2048-CPU Altix Bx2 system located at SGI's integration site in Chippewa Falls/Wisconsin.

The jump from the rather small 28- and 128-CPU Altix systems running at RRZE and LRZ today to the 2048-headed Hydra at the SGI site was an experience of its own. The machine was configured as four partitions of 512 CPUs each, with NUMALink4 connections in between. For all benchmarks, careful process placement was enforced using SGI's MPT (Message Passing Toolkit) facilities in order to prevent non-local memory access. On a production machine this task should of course be performed automatically by the batch system.

HHG Multigrid

The first application was an MPI-parallel multigrid finite element code that implements the HHG (Hierarchical Hybrid Grids) framework [1]. Every multigrid algorithm employs two components, a smoother for local error reduction and a coarse grid correction step. This code uses V-cycles for restriction to the co-arsest grid level and subsequent prolongation. Due to memory restrictions – the machine contained only 1 GB of memory per processor – only up to 1024 CPUs could be used, i.e. one CPU per node. Figure 1 shows performance per processor, broken down into smoothing and V-cycle parts. Up to 512 CPUs, weak scaling was used, i.e. the grid size was scaled proportionally to the number of processors. Taking the step to 1024 CPUs, however, the number of unknowns was held constant (strong scaling) to be $1.7 \cdot 10^{10}$. At this problem

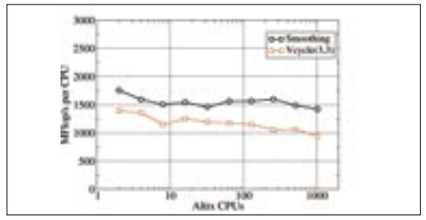


Figure 1: Parallel performance of smoothing and V-cycle implementations. Weak scaling up to 512 CPUs, strong scaling from 512 to 1024 CPUs

size, the wallclock time needed to perform ten V(3,3) cycles was well below one minute, with the application running at a sustained performance of roughly 1 TFlop/s. The results show that the parallel implementation is very efficient, and gives rise to the expectation that it should be possible to solve a problem with $6.8 \cdot 10^{10}$ unknowns on a machine with 4 GB per CPU (like HLRB 2). To our knowledge, even the memory-limited test case with $1.7 \cdot 10^{10}$ unknowns is the largest finite element system solved on a supercomputer to date. A report on this achievement has been accepted as student paper for SC 2005 [1].

BEST

The second benchmark program we put to the test was the well-known BEST code, a highly tuned MPI-parallel lattice Boltzmann implementation that was the subject of considerable optimization effort in the course of the last 12 months [2, 3, 4]. Performance of BEST is typically reported as MLup/s (Mega Lattice site updates per second). As a standard test case, an empty channel of size 100_128^2 per CPU was used (weak scaling). In BEST, the number of Flops performed per Lup varies due to different optimization strategies. We nevertheless report Flops along with Lup/s to give some impression on

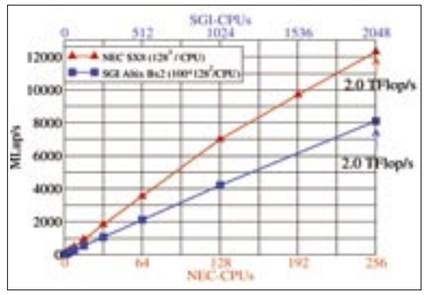


Figure 2: BEST performance and scalability for up to 2048 Altix CPUs. For comparison, NEC SX-8 data is also shown. Note that the ratio of MLup/MFlop differs between architectures

numerical efficiency. Figure 2 shows that a performance level of 2 TFlop/s can be reached on 2048 Altix CPUs, corresponding to roughly 8000 Mlup/s. For comparison, performance data on a NEC SX-8 machine at HLRS is included with a slightly larger domain size. The NEC SX-8 achieves the same performance as the full Altix machine at 160 CPUs. At 256 CPUs and 2 TFlop/s it peaks out at just over 12000 Mlup/s. It must be noted that BEST is not limited by memory bandwidth on modern vector processors but by the arithmetic units. On an Altix with its 0.06 Words/Flop balance, the code is certainly memory bound. Taking into account that every lattice site update requires 19 double precision loads and an equal number of stores, the full machine delivered an application bandwidth of 2.4 TByte/s. As each store miss leads to a read for ownership (RFO) of the corresponding cache line, the aggregated bandwidth through the memory buses was 3.6 TByte/s, more than half the theoretical limit. For reference, the official STREAM triad bandwidth on a 512-CPU Altix node is 1.1 TByte/s [5]. Due to the exceptional bandwidth of the NUMALink4 network, scalability is obviously not a problem.

In summary, we have benchmarked two highly scalable, memory-bound MPI applications on a 2048-processor Altix system. Both were shown to be capable of TFlop/s sustained performance and show very good scalability up to very large CPU numbers. They are thus among the "grand challenge" codes of choice for HLRB 2.

References

[1] **B. Bergen, F. Hülsemann, U. Rüde**
Is $1.7 \cdot 10^{10}$ unknowns the largest finite element system than can be solved today? Accepted as student paper for SC 2005

[2] **T. Pohl, F. Deserno, N. Thürey, U. Rüde, P. Lammers, G. Wellein, T. Zeiser**
Performance evaluation of parallel large-scale lattice Boltzmann applications on three supercomputing architectures. In: Proceedings of SC 2004. CD-ROM (2004)

[3] **G. Wellein, T. Zeiser, S. Donath, G. Hager**
On the single processor performance of simple lattice Boltzmann kernels. Submitted to Computers & Fluids (2005)

[4] **G. Wellein, T. Zeiser, P. Lammers, U. Küster**
Towards optimal performance for lattice Boltzmann applications on terascale computers. In: Parallel Computational Fluid Dynamics 2000, Trends and Applications. Proceedings of the Parallel CFD 2005 Conference, May 24-27, Washington D.C., USA. (2005) submitted

[5] **www.cs.virginia.edu/stream/tuned/Bandwidth.html**

- Georg Hager[1]
- Benjamin Bergen[2]
- Peters Lammers[3]
- Gerhard Wellein[1]

[1] Regionales Rechenzentrum Erlangen (RRZE)
[2] Los Alamos National Laboratory (LANL)
[3] High Performance Computing Center Stuttgart (HLRS)

Periscope: Performance Analysis on Large-Scale Systems

Introduction

There is a continuing trend in high performance computing (HPC) towards systems composed of an ever increasing number of processing elements. For example, the current leader of the Top 500 supercomputer list, a preliminary installation of IBM's BlueGene/L supercomputer, contains more than 30,000 processors. Several other HPC systems also consist of thousand or more processors. Such massive processor counts pose new challenges for application developers to obtain high performance for their codes. Intensive application tuning is often necessary to take full advantage of the capabilities a HPC system provides.

For application tuning the execution behavior on the target system must be monitored and analyzed. The analysis reveals information about inefficiencies in the application (performance bottlenecks) that can be used by the application programmer to improve the code. The size of the recorded history of a program's execution (the performance data) can easily be in the range of several gigabytes for large-scale systems. Since it is impossible to analyze such huge amounts of data manually, it becomes increasingly difficult to perform tuning efficiently (Figure 1). Approaches to automate the process of performance analysis on large systems are therefore needed. Here we present the Periscope environment, which leverages several new technologies to automatically detect performance bottlenecks.

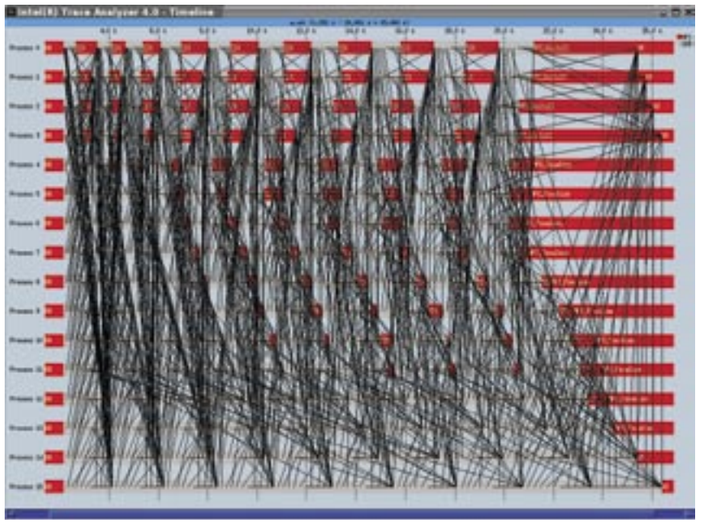


Figure 1: Manual performance analysis can be difficult and time-consuming

Periscope

Periscope is a DFG-funded project that continues work performed in the KONWIHR-funded Peridot project. In Peridot the goal was to investigate approaches for performance analysis for the Hitachi SR 8000 supercomputer at LRZ Munich which was the fastest supercomputer in Europe at that time and one of the first supercomputers with more than 1000 processors. In Peridot we devised a methodology for performance analysis for large-scale systems based on three main ideas:

- Automation: The search for a program's inefficiencies is conducted automatically. This can be achieved by formally stating what constitutes inefficient behavior in the form of performance property specifications. A language called ASL (APART Specification Language) was devised for this formal specification

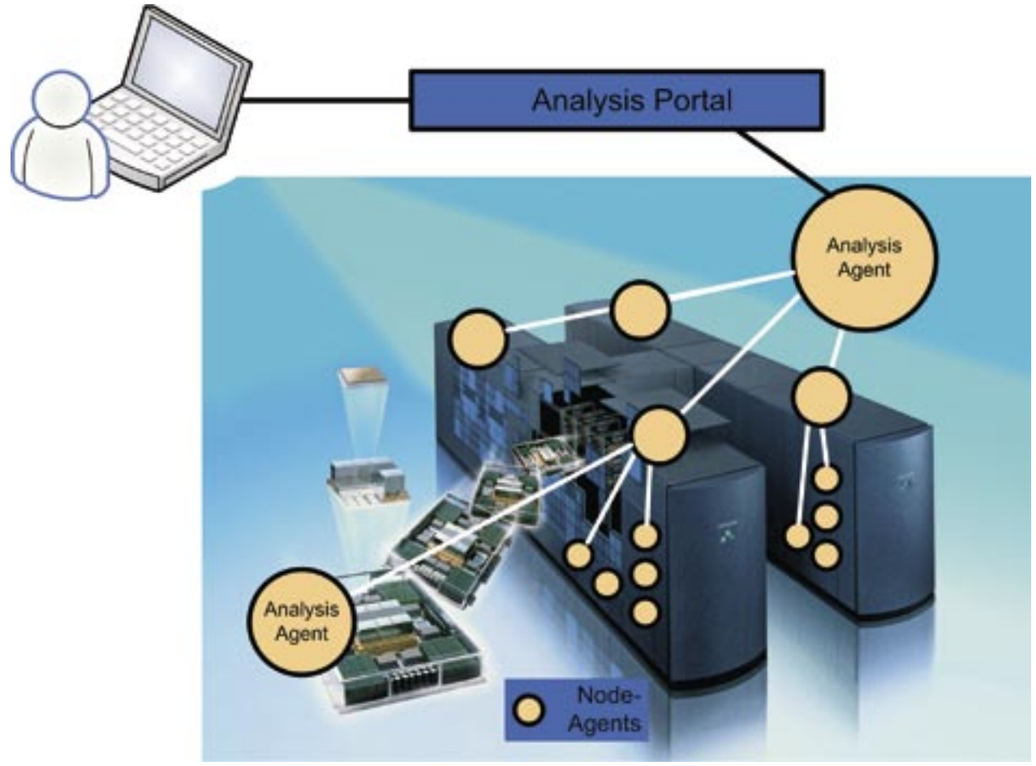


Figure 2: The Periscope performance analysis system is composed of a number of analysis agents arranged in a hierarchy

- Distribution: Instead of collecting and analyzing performance data centrally, in Periscope performance data are processed in a distributed fashion by a set of analysis agents

- Online operation: Instead of analyzing performance data after the program's execution, the analysis is performed while the application is running. This allows the search strategy to be adapted according to the current state of the analysis.

Performance Properties

The APART specification language allows the formal specification of performance properties. In addition to a condition that needs to be fulfilled for a property to hold, severity and confidence measures are included in the specification. The confidence value measures the level of confidence that a property holds. Severity measures the relative importance of the property, i.e., the amount of negative impact on the performance a property exhibits.

As an example consider the property "ImbalanceInParallelSection" that quantifies the work imbalance with respect to several threads executing a parallel section. The severity measure indicates the amount of time lost due to the imbalance situation.

Monitoring

A performance tool needs a way to obtain data about the target program's execution. We designed a novel

monitoring approach for Periscope that tries to leverage special hardware features of the SR 8000 supercomputer. The SR 8000 is a clustered SMP system with 8 processors per node available to applications. In addition there is a ninth processor per node (the service processor hosting the operating system). In our monitoring approach, performance data are written to shared memory regions by the processes of the target application. The performance data are then analyzed by an analysis agent that can reside on the service processor.

For other clustered SMP systems our monitoring approach can be used in a similar way. Either the agents are co-located with the target application or one or more processors are dedicated to performance analysis per node. Furthermore we can exploit remote direct memory access capabilities (RDMA) available on some modern architectures (e.g., Infiniband) to access a remote node's memory without the intervention of the processors of the remote node. A similar RDMA facility is also available on the SR 8000 (Figure 3, Figure 4).

Agents

Periscope is not a monolithic tool. Instead it consists of a number of agents that are distributed over the parallel machine (Figure 2). The agents co-operate in the detection of performance properties. On the lowest level there is one node-level agent per SMP node in the system. A node-level agent is responsible for the detection of node-local properties. For example, properties related to shared memory parallelism can be detected locally. Above the node level agents are arranged in a hierarchy. Depending on the size of the machine, the hierarchy may be shallow or deep. On the top of the hierarchy, a master agent is responsible for the interaction with the user. The master agent reports the detected properties to the user and takes possible steering commands as input. Intermediate agents are needed for the search for properties that cannot be detected locally because performance date from more than one node must be analyzed.

Status

We are currently in the process of creating a prototypical implementation of the Periscope system. The progress made so far indicates that Periscope can be a viable and efficient approach to performance analysis on large-scale systems.

Acknowledgements

The Peridot project was funded by was Competence Network for High Performance Computing in Bavaria KONWIHR (http://konwihr.in.tum.de) and by the European Commission via the APART working group (http://www.fz-juelich.de/apart). The follow-up project Periscope is funded by the Deutsche Forschungsgemeinschaft (DFG).

Periscope Project Homepage
www.lrr.in.tum.de/~gerndt/home/Research/PERISCOPE/Periscope.htm

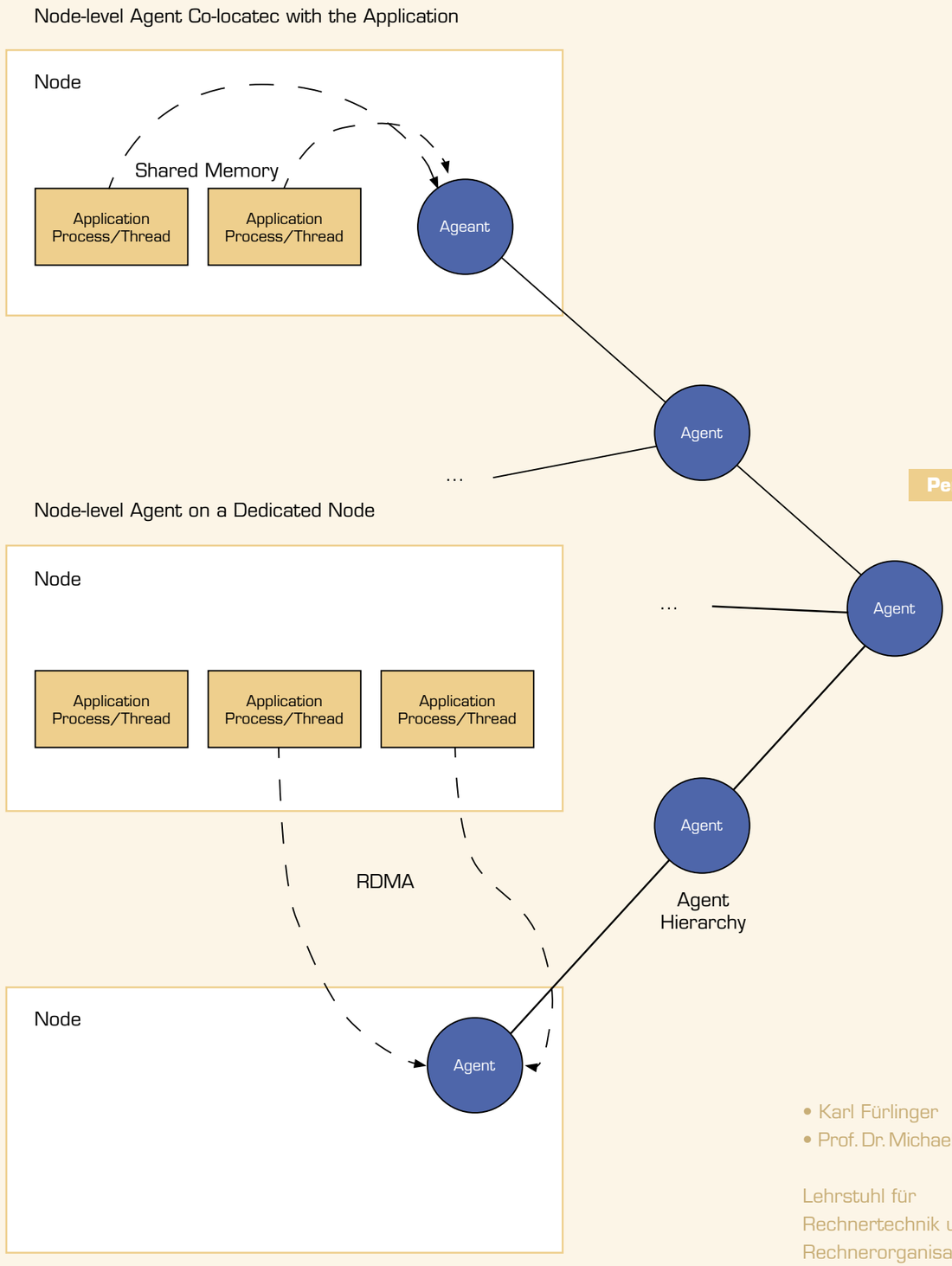


Figure 3: The Periscope monitoring approach relies on shared memory or uses an RDMA mechanism

• Karl Fürlinger
• Prof. Dr. Michael Gerndt

Lehrstuhl für Rechnertechnik und Rechnerorganisation
Technical University Munich

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 platform described below is documented on the LRZ WWW server; please choose the appropriate link from www.lrz.de/services/compute

Contact
High Performance Systems Department

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

View of the Hitachi SR8000-F1 at LRZ

Compute servers currently operated by LRZ are given in the following table

System	Size	Peak Performance (GFlop/s)	Purpose	User Community
SGI N.N. **(installation: 2006)**	5120 cores 25 TByte	33,000	Capability computing	German universities and research institutes
Hitachi SR 8000-F1 8-way	168 nodes 1344 processors (+168 Service procs) 1376 GByte memory	2,016	Capability computing	German universities and research institutes
SGI Altix 64-way (Q3 2005)	64 processors 256 Gbyte memory	410	Testing and porting	German universities and research institutes
SGI Altix 128-way	128 processors 512 Gbyte memory	820	Capacity computing	Bavarian universities
Linux Cluster Intel IA64 2-way	68 nodes 136 processors 816 GByte memory	870	Capability and capacity computing	Bavarian universities
Linux Cluster Intel IA64 4-way	17 nodes 68 processors 218 GByte memory	354	Capacity computing	Munich universities
Linux cluster Intel IA32 Intel&AMD EM64T	154 nodes 192 processors 320 GByte memory	850	Capacity computing	Munich universities
IBM pSeries 690 hpc 8-way	1 node 8 processors 32 GBytes memory	42	Capacity computing	Munich universities

A detailed description can be found on LRZ's web pages: www.lrz.de/services/compute

Based on a long tradition in supercomputing at Universität Stuttgart, 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.

Systems & Services
Operation of its systems is done together with T-Systems, T-Systems sfr, and Porsche in the public-private joint venture hww (Höchstleistungsrechner für Wissenschaft und Wirtschaft). Through this co-operation a variety of systems can be provided to its users.

Training & Education
HLRS runs eduaction and training courses at the University of Stuttgart, across Germany and with international partners worldwide. The course program covers all aspects of high performance computing. About 300 scientists receive training every year.

Centers of Excellence
In order to bundle service resources in the state of Baden-Württemberg HLRS has teamed up with the Computing Center of University of Karlsruhe and the University of Heidelberg in the Höchstleistungsrechner-Kompetenzzentrum Baden-Württemberg (www.hkz-bw.de). Together with another group of scientists called WIR (www.wir-bawue.de) HLRS works on research in scientific computing

Together with its partners HLRS provides the right architecture for the right application and can thus serve a wide range of fields and a variety of user groups.

View of the NEC SX-8 at HLRS

Compute servers currently operated by HLRS are

System	Size	Peak Performance (GFlop/s)	Purpose	User Community
NEC SX-8	72 8-way nodes 9,22 TByte memory	126,700	Capability computing	German universities, research institutes, and industry
TX-7	32 way node 256+512 GByte memory	192	Preprocessing	German universities, research institutes, and industry
Intel Nocona Cluster	205 2-way nodes 240 GByte memory	2,624	Capability computing	German universities, research institutes, and industry
Cray Opteron	129 2-way nodes 512 GByte memory	1,024	Capability computing	German universities, research institutes, and industry

Contact
Höchstleistungsrechenzentrum Stuttgart (HLRS)
Universität Stuttgart

Prof. Dr.-Ing. Michael M. Resch
Nobelstraße 19
70569 Stuttgart
Germany
Phone +49 711 685 872 69
resch@hlrs.de
www.hlrs.de

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. Its tasks are:

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 at DESY in 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, two research groups exist: the group Elementary Particle Physics, headed by Karl Jansen and located at the DESY laboratory in Zeuthen and the group Computational Biology and Biophysics, headed by Ulrich Hansmann at the Research Centre Jülich.

Education and training in the fields of supercomputing by symposia, workshops, school, seminars, courses, and guest programmes.

The IBM supercomputer "Jump" in Jülich (Photo: Research Centre Jülich)

The following supercomputers are available for research projects of the communities mentioned below, evaluated by the Peer Review Board of NIC. A more detailed description of the supercomputers can be found on the web servers of the Research Centre Jülich and of the German Electron Synchrotron DESY, respectively:

www.fz-juelich.de/zam/CompServ/services/sco.html
www-zeuthen.desy.de/main/html/home/

System	Size	Peak Performance (GFlop/s)	Purpose	User Community
IBM pSeries 690 Cluster 1600 "Jump"	41 SMP nodes 1312 processors POWER4+ 5248 GByte memory	9,000	Capability computing	German universities, research institutes, and industry
IBM Blue Gene/L "JUBL"	1 rack 1024 nodes 2048 processors PowerPC 440 512 GByte memory	5,600	Capability computing	German universities, research institutes, and industry
APEmille (special purpose computers)	4 racks 1024 processors 32 GByte memory	550	Capability computing	Lattice gauge theory groups at Universities and research institutes

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

Prof. Dr. Dr. Thomas Lippert
52425 Jülich
Germany
Phone +49 24 61 61 64 02
th.lippert@fz-juelich.de
www.fz-juelich.de/nic

High Performance Computing Courses and Tutorials

LRZ
www.lrz.de

Perspectives of High End Computing

Date
December 7, 2005
Location
Rechenzentrum Garching
Contents
- New application areas
- Planned applications and new developments
- Experiences with the migration system and current status of HLRB2
- Configuration and resource management of the new system
- Collaborations between research groups
- Requirements for support, software, tools and libraries
- Discussion with center staff about resource allocation and modes of operation.

Programming of High Performance Systems

Date
March 6-10, 2006
Location
Regionales Rechenzentrum Erlangen
Contents
- Basic concepts in HPC and modern HPC architectures
- Programming with MPI and OpenMP
- Optimization for modern processor architectures
- Itanium architecture, compilers, and tools
- SGI Altix architecture and tools
- Examples for the optimization of Fortran, C, C++, and I/O
- Tracing, profiling, threading and analysis tools.

This course is organized as a collaboration of LRZ, RRZE, Intel, and SGI.
Course language is English, if requested.

Iterative Linear Solvers and Parallelization

Date
October 9-13, 2006
Location
Garching, LRZ
Contents
The workshop is the same as the workshop held at HLRS in March 2006. The course is organized by LRZ, HLRS, University of Kassel, and IAG.
Webpage
www.hlrs.de/news-events/external-events/

HLRS
www.hlrs.de

Parallel Programming with MPI, OpenMP and PETSc

Date
March 1-3, 2006
Location
Dresden, ZIH
Contents
The focus is on programming models MPI, OpenMP, and PETSc. 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. This course is organized by ZIH in collaboration with HLRS.
Webpage
www.hlrs.de/news-events/external-events/

Iterative Linear Solvers and Parallelization

Date
March 13-17, 2006
Location
Stuttgart, HLRS
Contents
The focus is on iterative and parallel solvers, the parallel programming models MPI and OpenMP, and the parallel middleware PETSc. Different modern Krylov Subspace Methods (CG, GMRES, BiCGSTAB ...) as well as highly efficient preconditioning techniques are presented in the context of real life applications. Hands-on sessions (in C and Fortran) will allow users to immediately test and understand the basic constructs of iterative solvers, the Message Passing Interface (MPI) and the shared memory directives of OpenMP. This course is organized by University of Kassel, HLRS, and IAG.
Webpage
www.hlrs.de/news-events/events/

NEC SX-8 Usage and Programming

Date
March 20-21, 2006
Location
Stuttgart, HLRS
Contents
The first day is focused on vectorizing and parallelizing on NEC SX-8, the second day is dedicated to parallel I/O.
Webpage
www.hlrs.de/news-events/events

Introduction to Computational Fluids Dynamics

Date
April 3-7, 2006
Location
University of Kassel
Contents
Numerical methods to solve the equations of Fluid Dynamics are presented. The main focus is on explicit Finite Volume schemes for the compressible Euler equations. Hands-on sessions will manifest the content of the lectures. Participants will learn to implement the algorithms, but also to apply existing software and to interpret the solutions correctly. Methods and problems of parallelization are discussed. This course is organized by University of Kassel, HLRS, and IAG, and is based on a lecture and practical awarded with the "Landeslehrpreis Baden-Württemberg 2003" (held at University of Stuttgart).
Webpage
www.hlrs.de/news-events/external-events/

C++ for Scientific Computing

Date
April 3-13, 2006
Location
Stuttgart, HLRS
Contents
This introduction to C++ is taught with lectures and hands-on sessions. This course is organized by HLRS and Dept. for Computational Physics/University of Stuttgart.
Webpage
www.hlrs.de/news-events/events/

Parallel Programming with MPI, OpenMP and PETSc

Date
August 23-25, 2006
Location
Manno (CH), CSCS
Contents
The focus is on programming models MPI, OpenMP, and PETSc. 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. This course is organized by CSCS in collaboration with HLRS.
Webpage
www.hlrs.de/news-events/external-events/

Introduction to Computational Fluids Dynamics

Date
September 18-22, 2006
Location
Stuttgart, HLRS
Contents
The workshop is the same as the workshop held at HLRS in April 2006.
Webpage
www.hlrs.de/news-events/events

Parallel Programming with MPI, OpenMP and PETSc

Date
September 25-27, 2006
Location
Stuttgart, HLRS
Contents
The focus is on programming models MPI-1, OpenMP, and PETSc. It includes also an overview on MPI-2. 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. Course language is English if required.
Webpage
www.hlrs.de/news-events/events

Advanced Topics in Parallel Programming

Date
September 28-29, 2006
Location
Stuttgart, HLRS
Contents
Topics are MPI-2 parallel file I/O, OpenMP tools and tuning, hybrid mixed model MPI+OpenMP parallelization, domain decomposition of structured and unstructured grids and with particle based applications, parallel numerics, object oriented parallel programming with C++, and Grid computing. Course language is English if required.
Webpage
www.hlrs.de/news-events/events

NIC
www.fz-juelich.de/nic

Parallel Programming with MPI, OpenMP and PETSc

Date
November 28-30, 2005
Location
NIC/ZAM, Research Centre Jülich
Contents
The workshop is the same as the workshop held at HLRS in September 2006. This course is organized by NIC/ZAM in collaboration with HLRS. Presented by Dr. Rolf Rabenseifner/HLRS
Webpage
www.fz-juelich.de/zam/neues/termine/mpi-openmp

CECAM Tutorial Programming Parallel Computers

Date
January 23-27, 2006
Location
NIC/ZAM, Research Centre Jülich
Contents
This tutorial provides a thorough introduction to scientific parallel programming. It covers parallel programming with MPI and OpenMP. Lectures will alternate with hands-on exercises.
Webpage
www.cecam.fr/indexphp?content=activities/tutorial

Winter School Computational Nanoscience: Do it yourself

Date
February 14-22, 2006
Location
NIC/ZAM, Research Centre Jülich
Contents
The School focuses on the application of modern electronic structure calculations and dynamical simulation techniques. A practical introduction to the theory behind and handling of pertinent software packages through practicals and tutorials in small groups will be provided. The School is tailored for PhD students and Postdocs with a solid background in basic quantum mechanics coming from physics, chemistry, material science and related disciplines.
Webpage
www.fz-juelich.de/conference/wscn

Miscellany

Inauguration of New HLRS Building and the NEC SX-8/576M72

On July 21st and 22nd the new building of the High Performance Computing Center Stuttgart (HLRS) was inaugurated together with the new NEC SX-8/576M72 system. The president of the Universität Stuttgart Prof. Dr. Dieter Fritsch and the director of HLRS Prof. Dr. Michael Resch welcomed the Minister President of Baden-Württemberg Günther Oettinger and Federal Minister of Education and Research Edelgard Bulmahn. Their presence at the inauguration made clear the importance of the national supercomputing center HLRS both for the state of Baden-Württemberg and Germany. Consequently both Federal Minister Bulmahn and Minister President Oettinger emphasized the close collaboration of state and federal government in pushing the frontiers of science and research by supporting large scale infrastructure like the new NEC SX-8 system. Minister President Oettinger pointed out the importance of HLRS for the state of Baden-Württemberg both for research and for industry. He made it clear that becoming a European supercomputing center and keeping its role as a leading partner and provider for industry should be a key goal for HLRS for the next years.

The importance of the installation at HLRS was emphasized also by the attendance of Senior Vice President Masahiko Yamamoto from NEC Corporation, Tokyo. Senior Vice President Yamamoto brought the greetings of NEC president Akinobu Kanasugi and pointed out that the importance of HLRS and the SX-8 installation for NEC is reflected in the close collaboration of HLRS and NEC leading to additional

investment of about 8 MEuro by NEC in Stuttgart in the coming four years.

Dr. Christoph Gümbel head of simulation and computation at Porsche Research brought greetings from Porsche AG which is one of the main users of the systems at HLRS. Dr. Gümbel emphasized the importance of high speed computers for industry and expressed his hope that the very positive collaboration with HLRS will be continued and extended in the future. In a press release the CEO of Porsche AG Dr. Wendelin Wiedeking was quoted as “Commerce and science in Germany have a shared interest in powerful supercomputers,

which are becoming more and more important for research and development in the face of international competition”. According to Wiedeking, the many years of commitment being provided by the automobile manufacturer would contribute not only to improved levels of efficiency in development work at Porsche, but would also ultimately strengthen Germany’s standing in the world of science.

On July 22nd a scientific colloquium was held showing the variety of applications users of the HLRS are working on. The day was opened by a key note speech of Dr. Ryutaro Himeno head of the computer and information division of the Japanese Center for Physical and Chemical Research (RIKEN). Dr. Himeno pointed out the achievements of the last two decades in supercomputing he emphasized that the community should prepare for similar achievements in technology for the next years. As an example Dr. Himeno presented the Japanese Next Generation Supercomputing project for the development of a Petaflop computing system. HLRS and RIKEN will collaborate in this project focusing on application and system design issues.

In the following a number of brilliant presentations gave a glimpse of what will be possible in the future with the new NEC SX-8/576M72 system. Prof. Wolfgang Wall chair for Computational Mechanics of the Technical University of Munich presented some projects in multiphysics applications. It became obvious that this filed will become increasingly important and will require a tremendous level of performance. Prof. Cord Rossow director of the Institute for Design Aerodynamics of the German Aerospace Center (DLR) gave an overview of supercomputing in aerospatial research. Dr. Matthias Meinke from the Institute of Aerodynamics of the RWTH Aachen gave a comprehensive overview

of new challenges in aerodynamics simulation. Both Prof. Rossow and Dr. Meinke made clear that supercomputers will remain a key tool for research and development in aerospatial engineering for decades to come.

Prof. Hanns Ruder from the Institute for Astronomy and Astrophysics (IAAT) of the Eberhard-Karls-Universität Tübingen showed the potential of supercomputers in visualization. His presentation of the effects of traveling at a speed close to the speed of light was very impressive for the audience. A similar approach of using visualization coupled to supercomputer simulations was presented by Prof. Eberhard Göde from the Institute for Fluid Mechanics and Hydraulic Machinery of Stuttgart University.

Munich Centre for Computational Sciences founded

After the move of LRZ to the Campus Garching in spring 2006 two major German supercomputing centres will be located within a distance of a few hundred metres: Leibniz Computing Centre (LRZ) and the Computing Centre of the Max-Planck Society (RZG). This opens up numerous new opportunities for further development of the supercomputing infrastructure in Germany.

Signing the collaboration agreement on July 18, 2005 (from left to right): S. Heinzel (Director of RZG), Prof. Herrmann (President of Technical University), Prof. Nöth (President of the Bavarian Acadamy of Sciences), Prof. Gruss (President of Max-Planck Society), Prof. Huber (Rector of Ludwig-Maximilians-Universität), Prof. Hegering (Director of LRZ)

On July 18, 2005, leading representatives of the Max-Planck Society, RZG, the Universities of Munich (Ludwig-Maximilians-Universität and Technical University), the Bavarian Academy of Sciences and LRZ have put their signature on a far-reaching agreement for collaboration in the area of Computational Sciences. Based on the ideal combination of HPC expertise and supercomputing capabilities contributed by RZG and LRZ, educational and research facilities provided by the universities, and interdisciplinary as well as fundamental research undertaken by the Max-Planck Society this collaboration intends to

- support the development of algorithms and applications suitable for high performance computing, as well as data processing and visualization in various scientific areas

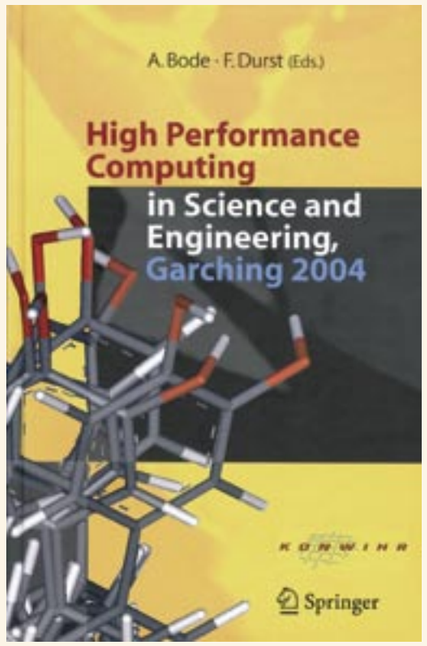
- bundle the two supercomputing centres' competence in the areas of high performance computing, large scale data handling and archiving, high performance networking, and optimization of HPC applications

and to thereby provide a quality of scientific computing unique throughout Germany, and on an internationally competitive level. The agreement is prompted by the increasing demands of the leading research establishment in or near Munich for HPC capacity and expertise.

Springer: Transactions of KONWIHR

This volume of the series High Performance Computing in Science and Engineering is fully dedicated to the final report of KONWIHR, the Bavarian Competence Network for Technical and Scientific High Performance Computing. It includes the transactions of the final KONWIHR Workshop, that was held at Technische Universität München, October 14-15, 2004, as well as additional reports of KONWIHR research groups. KONWIHR was established by the Bavarian State Government in order to support the broad application of high performance computing in science and technology throughout the country. KONWIHR is a supporting action to the installation of the German supercomputer Hitachi SR 8000 at the Leibniz Computing Center of the Bavarian Academy of Sciences.

The report covers projects from basic research in computer science to develop tools for high performance computing as well as applications from Biology, Chemistry, Electrical Engineering, Geology, Mathematics, Physics, Computational Fluid Dynamics, Materials Science, and Computer Science.



High Performance Computing in Science and Engineering, Garching 2004
Transaction of the KONWIHR Result Workshop, October 14-15, 2004
Technical University of Munich, Garching, Germany
Bode, Arndt; Durst, Franz (Eds.) 2005, XII, 301 p. 190 illus., 13 in colour, Hardcover
ISBN: 3-540-26145-1

The 2nd Russian-German School on Parallel Programming

using
High Performance Computation Systems
June 27- July 8, 2005
Novosibirsk/Russia

The present 2nd School was the continuation of the successful 1st School (July, 5-9, 2004, Novosibirsk/Russia) and was devoted to the problems of parallel computations using modern high performance computing systems.

The duration of the School and the amount of the presented material have been doubled, that have allowed to discuss lot of important issues in detail. Additional topics like Single Processor Optimization, Parallel Debugging, Visualization Tools have been presented. In addition, the extended time of the School has allowed for extended practical sessions. Interesting and highly professional lectures have been given by T. Bönisch, R. Keller, B. Krammer. Russian scientists: Prof. Yu. Shokin, Prof. A. Fedotov, Prof. V. Khoroshevskii have given two review lectures on

telecommunicational resources of Siberian Branch of Russian Academy of Sciences and Russian investigations in the field of distributed systems and parallel computations.

The number of participants and their level of knowledge have increased. The 2nd School included participants from all scientific center of Siberian Branch of Russian Academy of Sciences: Novosibirsk, Tomsk, Kemerovo, Krasnoyarsk, Irkutsk, and also scientists from Kazakhstan: Almaty, Ust-Kamenogorsk.

The book of course material (261 pages) has been prepared by T. Bönisch, R. Keller, B. Krammer, printed out in ICT SB RAS and it has been given to every participant.

The conference room and PC pool have been specially equipped and prepared for the School.

Practical sessions were done on the cluster of ICT SB RAS, which is more powerful than the cluster of Novosibirsk State University, used for the 1st School. Besides, participants will use the cluster of ICT in the future, applying the knowledge learned from the School.

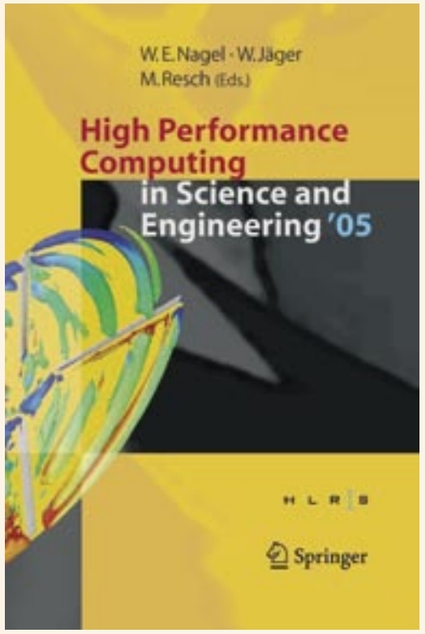
Real problems (molecular dynamics, CFD and combustion), provided by participants, were discussed in order to show the possible improvements in parallel programming.

After the visit of German scientists (Prof. M. Resch, U. Küster, Dr. N. Shokina) to Kazakhstan (International Conference “Computational and Informational Technologies for Research, Engineering and Education”, Almaty, October 6-10, 2004) and the initialization of international German-Kazakh cooperation, Kazakh scientists took part in the 2nd School.

The success of the 2nd Russian-German School on Parallel Programming using High Performance Computing Systems (RGSPPHPC 2005) shows the necessity of continuation of such events in the future.

The 2nd Russian-German School on Parallel Programming using High Performance Computing Systems has been supported by HLRS, ICT SB RAS, Russian Foundation for Basic Research and Siberian Branch of Russian Academy of Sciences.

Scientific Secretary of German-Russian Center for Computational Technologies and High Performance Computing



Springer: Transactions of the HLRS

HLRS held its 8th Results and Review Workshop on October 13-14, 2005 at Stuttgart. 45 scientists from all over Germany gave presentations on computational engineering and science. The best papers of this workshop are published as Transactions of the High Performance Computing Center Stuttgart (HLRS) by Springer as “High Performance Computing in Science and Engineering 2005”.

inSiDE

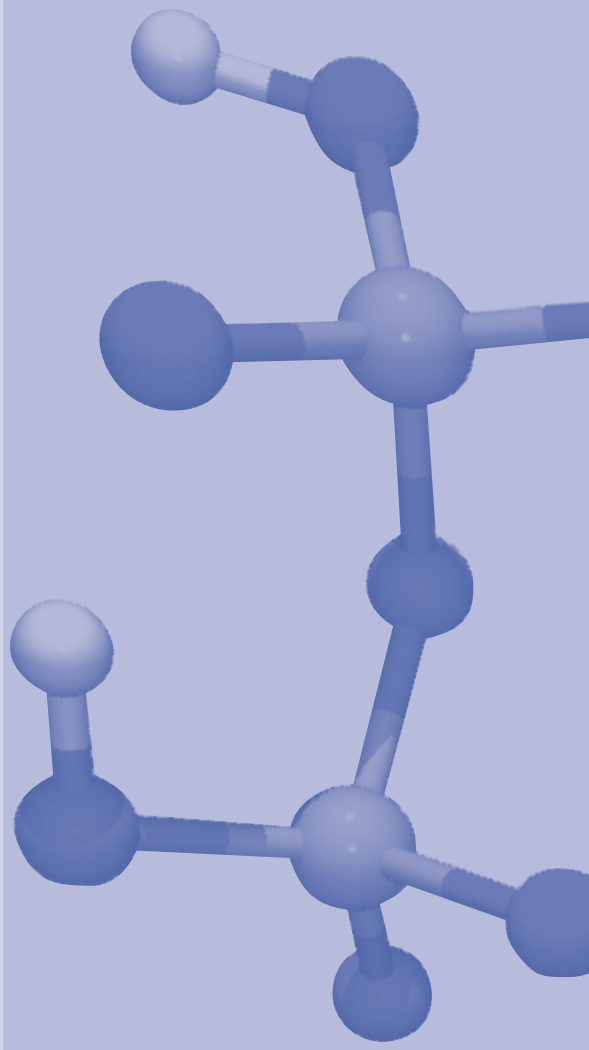
inSiDE is published two times a year by
The German National Supercomputing
Centres HLRS, LRZ, and NIC

Publishers
Prof. Dr. Heinz-Gerd Hegering, LRZ
Prof. Dr. Dr. Thomas Lippert, NIC
Prof. Dr. Michael M. Resch, HLRS

Editor
F. Rainer Klank, HLRS
klank@hlrs.de

Design
Katharina Schlatterer
kschlatterer@hlrs.de

Authors
Norbert Attig
n.attig@fz-juelich.de
Reinhold Bader
reinhold.bader@lrz-muenchen.de
Martin Becker
becker@hlrs.de
Benjamin Bergen
bergen@lanl.gov
Matthias Brehm
matthias.brehm@lrz-muenchen.de
Ralf Ebner
ralf.ebner@lrz-muenchen.de
Karl Fürlinger
karl.fuerlinger@fz-juelich.de
Michael Gerndt
m.gerndt@in.tum.de
Stefan Gottlöber
sgottloeber@aip.de
Georg Hager
georg.hager@rrze.uni-erlangen.de
Ulrich H. E. Hansmann
u.hansmann@fz-juelich.de
Arman Khalatyan
akhalatyan@aip.de
Anatoly Klypin
aklypin@nmsu.edu
Peter Lammers
plammers@hlrs.de
Gerhard Wellein
gerhard.wellein@rrze.uni-erlangen.de
Klaus Wolkersdorfer
k.wolkersdorfer@fz-juelich.de



If you would like to receive inSiDE regularly, send an email with your postal address to klank@hlrs.de

© HLRS 2005