

The background of the entire page is a close-up, macro photograph of numerous water droplets. The droplets are of various sizes and are highly reflective, creating bright highlights and dark shadows that give them a three-dimensional appearance. They are densely packed, filling the entire frame.

CREST 

NEWSLETTER
AUTUMN 2013

WELCOME

Welcome to this edition of the CRESTA Newsletter. Time is passing quickly within the project and it's hard to believe we've already completed 24 of our 36 months. We're learning an incredible amount about the challenges we face moving from the petascale to the exascale.

Although CRESTA focuses on the software challenges of exascale, we keep a keen eye on developments in the hardware arena. Recent comments from Intel's Chief Architect for Exascale Systems, Al Gara, show that we are not yet close to solving the power challenge - and that many of the probable solutions indicate more parallelism per flop than we have today. A key question is whether or not the new memory technologies we need, will appear in time. With many applications already struggling to scale, the levels of parallelism we expect at the exascale - certainly more than 100 million threads - are an enormous challenge for all of today's supercomputing applications.

CRESTA's approach from the outset has been to look at the sort of real-world applications that are in common use today and identify if they can continue to exist in something similar to their present form at the exascale or, if major disruptive changes to these codes will be required. As you will find out in this issue of the CRESTA Newsletter, this is a complex task where there is no single correct answer. We know that many of the applications we're looking at will not deliver a significant fraction of an exaflop in their current form. Understanding the balance between incremental and disruptive change to these applications has been key part of our recent CRESTA work.

I hope that you enjoy this issue of the CRESTA Newsletter!

Dr Mark Parsons
CRESTA Project Coordinator



CONTENTS

CRESTA Collaboration Meeting

4

CRESTA, Deep And Mont-Blanc

5

Vampir Takes a Bite on GROMACS

6

Efficient MPI Intercommunication

8

Elmfire: A Number Cruncher for Turbulence in Fusion Plasma

9

Nek5000 within CRESTA

10

OpenFOAM

12

CRESTA at SC13

13

Other Events

14

PGAS@EPCC

EASC2013

CRESTA on Screen

15

CRESTA COLLABORATION MEETING

MIKKO BYCKLING, CSC

At the end of September 2013, CSC organised the fifth CRESTA collaboration meeting in Espoo, Finland. The main focus of the meeting were emerging accelerator (General-Purpose Graphics Processing Unit, GPGPU) and coprocessor (Intel Many Integrated Core, MIC) technologies. In many ways these technologies, which are available already today, include many aspects which will be likely to be present in the first exascale computers of tomorrow, i.e., greatly increased parallelism and somewhat more complicated programming models.

The goal of the meeting was to bring together experts from different fields to share experiences and solve problems related to the efficient use of the accelerator and coprocessor technologies. Apart from the co-design aspect, the aim was also to give all participants the chance to gain first-hand experience on the efficient use of these technologies.

The main topic on the first day of the meeting was the Intel MIC architecture, i.e. Intel Xeon Phi. During the day, several partners gave presentations. CSC gave two presentations: the first on the Xeon Phi architecture and the second on programming the Xeon Phi. The

latter session was supported by a hands-on session on a local Xeon Phi cluster. The day then continued with presentations by CSC and KTH on porting real-world applications, IFS and GROMACS, to run on Xeon Phi. Finally, Allinea concluded the day by giving a presentation about debugging and performance profiling on the Xeon Phi.

The main topics of the second day were GPGPUs and the OpenACC programming model. The day started with EPCC giving a presentation about the OpenACC programming model, followed by a real-world case study regarding the HIMENO benchmark. The day continued with a presentation given by ABO about the productivity versus performance aspects of the OpenACC programming model. Finally, KTH presented the status of Nek5000 code, which has been enabled to use GPGPUs with OpenACC.

It is safe to say that the goals of the meeting were successfully met. During the meeting it became clear that to fully benefit from the new technologies, disruptive changes are needed. On the other hand, the programming environments for both Xeon Phi and GPGPUs have matured enough in order to enable high productivity without sacrificing too much performance.

Within CRESTA, the development of the co-design vehicles towards the exascale continues.

THE PARTNERS

Åbo Akademi University, Finland

Allinea Software, UK

Cray UK Limited, UK

Ecole Centrale Paris, France

EPCC, The University of Edinburgh, UK

German Aerospace Centre (DLR), Germany

Kungliga Tekniska högskolan (KTH), Sweden

The European Centre for Medium-Range Weather Forecasts, International

The University of Jyväskylä, Finland

The University of Stuttgart, Germany

Tieteen Tietotekniikan Keskus OY (CSC)'s IT Centre for Science Ltd, Finland

University College London, UK

ZIH, Technische Universität Dresden, Germany

CRESTA, DEEP AND MONT-BLANC

LORNA SMITH, EPCC

CRESTA is one of three projects which was funded by the EC in 2011 to lead Europe's strategy for producing European exascale hardware, software and applications. The other two projects, DEEP and Mont-Blanc are developing prototype hardware and are complementary to CRESTA's development of software and applications.

The three projects have always collaborated over dissemination activities, however, over the past year this collaboration has expanded and so Mont-Blanc hosted a joint collaboration meeting between the three projects in Barcelona during the summer. The meeting brought together many of the technical people working on the projects, and we had a series of talks describing the work of each.

In particular, we explored the different applications in each project and the challenges they face in preparing to utilise exascale platforms. We also considered the programming models and languages being used, looking at both the advantages and disadvantages of each approach. Finally, we explored the range of tools, from performance analysis, through debugging, to the pre- and post- processing tools, within CRESTA.

The result was a series of possible areas of overlap and potential collaborative opportunities. These include possible shared hands-on tools workshops, testing of programming models on each other's applications, and possible collaboration over workflows, data analysis, and visualisation. All in all, this was a very successful meeting with lots of opportunity to forge closer links. We are planning to hold a second meeting in Edinburgh in March of next year in order to build on this work.



VAMPIR TAKES A BITE ON GROMACS

HOLGER BRUNST, JENS DOLESCHAL, ANDREAS KNÜPFER, TUD

High Performance Computing (HPC) is obsessed with maximum computational performance and highest scalability pretty much by definition. The key ingredients to reach these goals besides very potent hardware platforms are innovative and appropriate parallel programming models, well-designed parallel application codes with ambitious development teams, and tailored software tools for the performance engineering process.

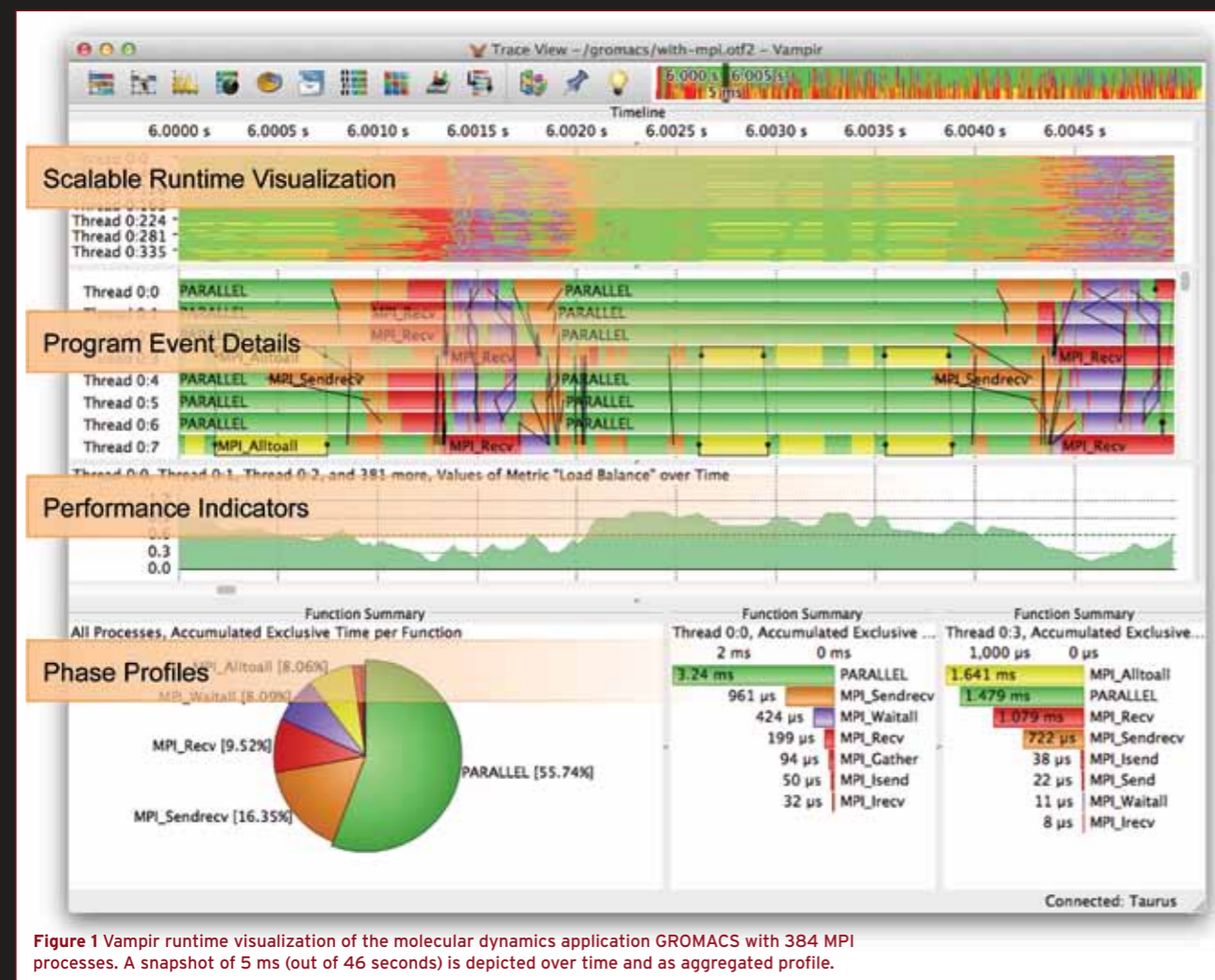


Figure 1 Vampir runtime visualization of the molecular dynamics application GROMACS with 384 MPI processes. A snapshot of 5 ms (out of 46 seconds) is depicted over time and as aggregated profile.

Whenever parallel performance and efficiency is important, their measurement should receive adequate attention. Measurements should provide sufficient detail, avoiding overwhelming perturbation at the same time. Performance should never be measured in a do-it-yourself manner comparable to “printf debugging”, because this is bound to be less reliable, more costly in terms of development and maintenance, and limited in scope and scalability.

Instead, dedicated software tools should be used, which automate all or most steps of the monitoring and analysis and which pay attention to all aspects of the parallel performance including the sequential

performance during independent parallel phases, the communication and synchronization, idle phases and waiting time, detailed information for selected phases in contrast to just global averages, hardware performance counters, synchronized measurement timing, and many more.

Vampir together with its runtime monitoring components is such a dedicated tool. Some of its most prominent features are highlighted in this article and illustrated with examples from the molecular dynamics application GROMACS, one of the six co-design applications in the CRESTA project.

MPI is the de-facto standard in HPC for distributed-memory parallel programming. Vampir started out more than 15 years ago as an MPI-centric tool as the letters “mpi” in “Vampir” tell. The initial versions of GROMACS up to version 4.5 were using MPI as sole parallelization. Figure 1 shows a characteristic Vampir visualization of an MPI-only run of GROMACS with 384 processors - the number of processes mainly depends on the input data set. Note the function/task-decomposition, which is typical for GROMACS where 3 out of 4 ranks compute the particle-particle interaction, while the others compute the Particle Mesh Ewald method (see Program Event Details).

Subsequently, the hybrid MPI-OpenMP combination has been the state-of-the-art for a number of years when adapting parallel codes to hierarchical platforms. OpenMP takes advantage of the shared-memory

parallelism inside the compute nodes while MPI performs the inter-node communication. Vampir supports this hybrid model just as well, showing MPI operations, OpenMP regions, and their interactions. In Figure 2, an example run of a hybrid MPI/OpenMP version of GROMACS is shown. The OpenMP threads are visible next to their master thread, which also acts as an MPI rank. The different communication phases for the 3D FFT as well as the computation phases are clearly visible.

Today, accelerator devices play an important role in HPC. Consequently, the GROMACS was extended with CUDA support in order to harness their potential for even faster computing. Likewise, Vampir provides analysis support for accelerator programming via Nvidia's CUDA and OpenCL as well as their combinations with MPI and OpenMP. Figure 3 shows an example of the Vampir visualization with CUDA API calls on the host side, the CUDA kernel invocations on the devices, and the memory transfers between both sides.

For application developers it is critical that all relevant parallel paradigms are covered under one hood, in order to tackle the tuning and scaling challenges of their specific parallel code. Accordingly, the Vampir team currently introduces support for PGAS parallelization

models like OpenSHMEM and GASPI as well as OpenACC. They will become available in 2014.

Besides the supported platforms and paradigms, scalability is the next important prerequisite for an ambitious software tool like Vampir. This concerns both runtime monitoring components and interactive visualization components. These two have been demonstrated with over 200 000 parallel processes and the Vampir team is eager to closely follow the levels of scalability of the world's leading supercomputers.

Vampir has always been strongly committed to parallel performance and scalability. For some time, energy consumption is having a growing impact on the HPC community. And obviously, the computing performance and the energy consumption should never be optimized separately, because of their non-trivial interplay. Therefore, Vampir can incorporate additional indicators like the power consumption measured with node-external power meters or internal CPU alternatives. They can be combined with typical performance metrics like the ones from PAPI. Figure 4 shows GROMACS with added power information.

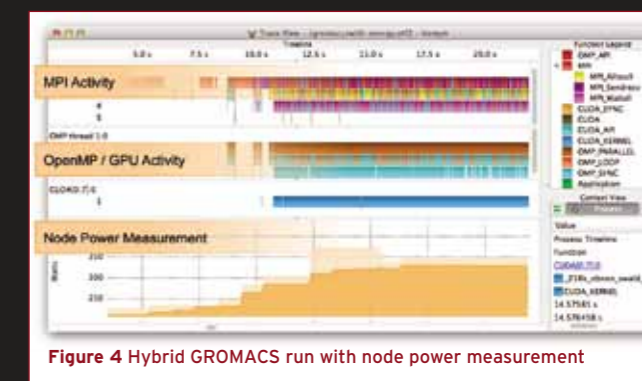


Figure 4 Hybrid GROMACS run with node power measurement

This short tour showed some of the most important features of Vampir, which is installed and ready for use at many of the leading HPC centers worldwide. For more information, see <http://www.vampir.eu> and <http://www.score-p.org/> or visit the ZIH booth #3905 at SC13.

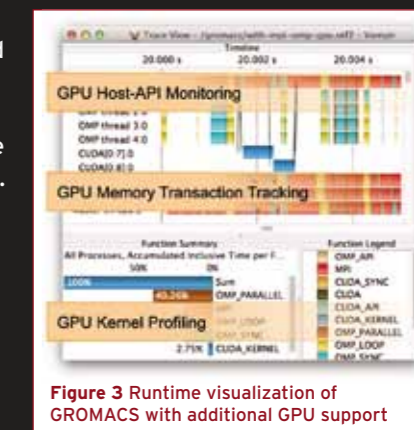


Figure 3 Runtime visualization of GROMACS with additional GPU support

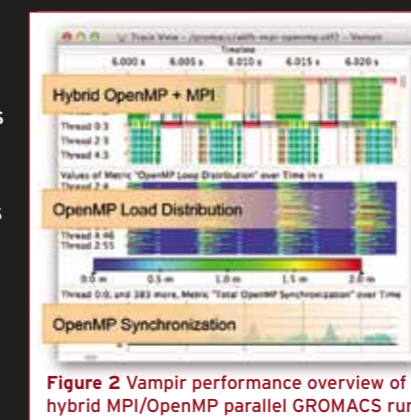


Figure 2 Vampir performance overview of hybrid MPI/OpenMP parallel GROMACS run.

EFFICIENT MPI INTERCOMMUNICATION

TUOMAS PUURTINEN, KEIJO MATTILA,
JUKKA TOIVANEN AND JUSSI TIMONEN, JYU

MPI intercommunication provides an efficient way to couple MPI-parallelised codes.

In many cases complexity and size of the simulation system is markedly increased by adding to it a new feature. It may not be simple to add it to the original code, and a solution to this problem is to have a separate code for the new feature and then run it in parallel with the original one. This can also lead to a very large simulation system whose execution may demand hardware that approaches even exascale. The scalability of such coupling of different pieces of software becomes then an issue, in particular when the original and additional codes are based on localized algorithms in which communication overheads are minimized. In the lattice-Boltzmann (LB) co-design group we are therefore developing a fully local method for coupling of LB codes of good scalability, which would linearly benefit from an increasing number of cores. The LB method uses a grid (lattice) to describe the fluid and solid regions so that it is fairly easy to use for example a tomographic image as the simulation geometry. For each lattice site variables are updated based on data from predefined neighbouring sites, which in the best case leads to an $O(N^3)$ over $O(N^2)$ cost of the computation and communication, respectively. In the CRESTA project the LB code to be developed is the HemeLB code of UCL. It models blood flow in human vasculature in the vicinity of, for instance, a cranial aneurysm, with the ultimate goal of providing data for clinical decision-making. The model would be naturally extended so as to involve colloidal drug particles which can diffuse from blood flow to the surrounding tissue of interest. Their diffusion in that tissue can best be described by an independent (LB) code. The two codes then need to be coupled as described above. The coupling interface to be developed should however be generic so that it could be used for coupling of any codes. We demand in addition that the changes its use demands from the two codes should be very small. To this end, instead of adopting an existing external coupler such as OASIS, we directly used the MPI intercommunication framework for a simple and efficient coupling between the solvers.

We tested the scaling of the MPI intercommunication interface by coupling in that way two LB solvers, first of which simulated blood flow with tracer particles that can penetrate the wall of the blood vessels to the surrounding tissue, and the second one then simulated their subsequent diffusion in that tissue. We found that the coupling of these codes had a negligible effect on their efficiency in a simulation geometry with an equal load balance on a 'small' number of cores (< 4000). Even though additional measurements of the performance must be done, the first results on scaling were promising. They indicate that this kind of coupling interface can effectively be used in very large simulations.

We have also looked at other possible applications of the MPI intercommunication interface. Even though LB scales well, it may not ideal to use the best possible resolution in the whole human vasculature as the actual regions of interest in the planned applications of HemeLB are spatially restricted. High grid density would generate data with unnecessarily details, thus increasing the amount of input and output data, and causing a deprivation of the overall efficiency. We are now investigating if the coupling interface could as well be used to couple different LB grids, and eventually also an LB code with a lower dimensional model for the regions of less interest.

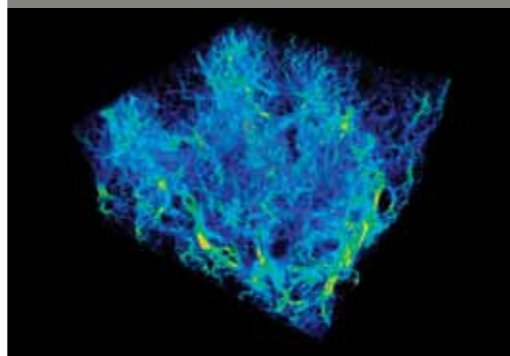


Figure 1 Flow velocity field as determined by LB simulation in a piece of vasculature generated from X-ray tomography data at the University of Jyväskylä

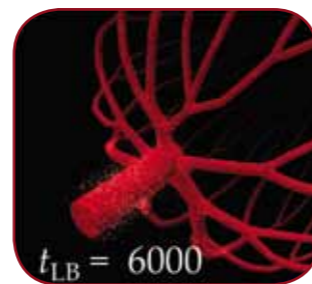
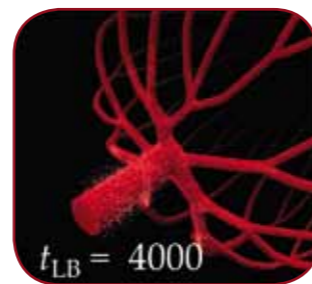
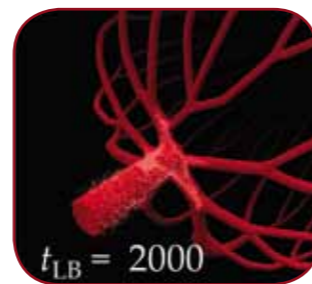
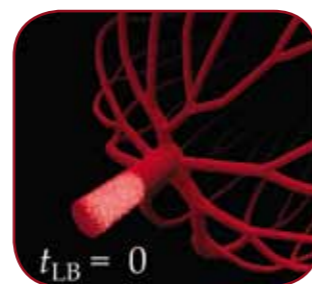


Figure 2 An (LB) advection-diffusion model for blood flow with tracer particles which can diffuse through the wall of (schematic) blood vessels.

ELMFIRE: A NUMBER CRUNCHER FOR TURBULENCE IN FUSION PLASMAS

JUKKA HEIKKINEN, TIMO KIVINIEMI, TUOMAS KORPILO AND SUSAN LEERINK,
VTT TECHNICAL RESEARCH CENTRE OF FINLAND; ARTUR SIGNELL AND JAN WESTERHOLM, ABO

Fusion energy is considered as an important future source for reliable, sustainable base load power. To address its demanding and costly science and technology challenges, fusion energy is being developed today with a concerted world-wide co-operation. Fusion energy uses plasma as a fuel. Plasma is the fourth state of matter and exists at very high temperatures. Its physics is very complex and its modelling for fusion reactors needs high performance computing resources. Today, with current supercomputing resources, it is impossible to perform first principles physics calculations of fusion plasma discharges in reactors. The major reason for this is the small-scale plasma turbulence structures affecting the plasma confinement. Exascale supercomputers could make these calculations feasible.

The Aalto University, ÅA, and VTT transport team have taken an initiative to develop a kinetic simulation code ELMFIRE for studying turbulence and transport in toroidal magnetized plasmas [1]. ELMFIRE is a global full particle distribution function (f) particle gyrokinetic¹ code. The particle orbits are solved in time in a 5D phase space, and a 3D electrostatic potential solver is included (see the figure below) to capture turbulence that arises. ELMFIRE is compatible with strong variations of the bulk plasma distributions with a variety of heat sources and sinks. The need to model full 3D electrostatic (or electromagnetic) potential together with particle motion to capture the turbulence was identified. This pushed the full f ELMFIRE project to start up.

Within the EU CRESTA project, the ELMFIRE code is provided with a 3D Domain Decomposition (DD) feature, among others, which is of paramount importance for enhanced memory usage of ELMFIRE. As the polarization drift is explicitly solved and the polarization density and parallel nonlinearities are implicitly solved in the gyrokinetic Poisson equation, the 3D DD has a strong impact on the memory usage of the coefficient matrix of the gyrokinetic Poisson equation.

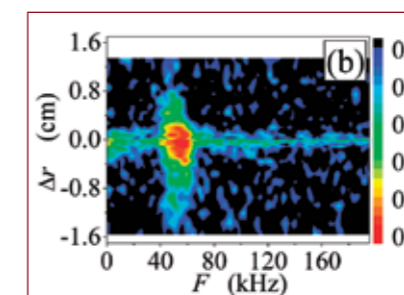
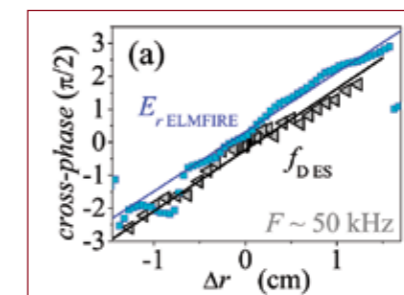
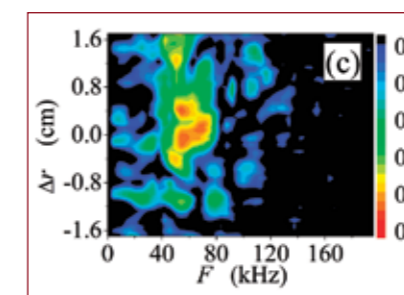


Figure (a) Cross-phase via radial separation showing agreement between GAM speed from ELMFIRE and FT-2 measurement. Coherence via spatial separation. (b) experimental for f_D ; (c) simulated for E_r .



¹ In gyrokinetic equations, an appropriate particle gyrophase averaging is done analytically thus reducing the phase space dimensions needed to be numerically resolved by one.

As the computing time and required memory with ELMFIRE grows with the size of the grid (grid cell size bound to ion Larmor radius) and with the number of simulation particles, the simulation requires the larger computing resources the larger the simulated plasma device is. For a reasonable simulation time scale for a JET (the largest tokamak in the world presently) plasma, ELMFIRE requires exascale supercomputing power.

After code optimizations performed at Aalto and at Åbo Akademi (supported by the Academy of Finland, by CSC, TEKES, and EU), ELMFIRE can be run with 32 000 cores in PRACE's Curie supercomputer for a $100+600+32$ grid using 3000 particles per cell, acceptable for turbulence saturation studies in reasonably sized volumes inside the ASDEX Upgrade (Garching, Germany) plasma. This will further improve with the introduction of the new features under development within the CRESTA project. More information can be found in the full article on the CRESTA website www.cresta-project.eu.

[1] <http://www.csc.fi/english/csc/publications/cscnews/2012/1>, "Simulating fusion reaction", CSC News 1 (2012) 4; J.A. Heikkinen et al., "Gyrokinetic simulation particle and heat transport in the presence of wide orbits and strong profile variations in the edge plasma", Contrib. Plasma Physics 46 (2006) 490.

NEK5000 WITHIN CRESTA

CRESTA WP6 Nek5000 team: Dan Henningson, Philipp Schlatter, Adam Peplinski (KTH Mechanics), Stefano Markidis, Michael Schliephake, Erwin Laure, Jing Gong (PDC, KTH), Alistair Hart (Cray UK), David Henty (EPCC), Paul Fischer and Katherine Heisey (ANL)

Nek5000 is an open-source code for simulating incompressible flows [1]. The Nek5000 discretization scheme is based on the spectral-element method. In this approach, the incompressible Navier-Stokes equations are discretized in space by using high-order weighted residual techniques employing tensor-product polynomial bases. The resultant linear systems are computed using Conjugate Gradient iteration (CG) with convenient pre-conditioners. The tensor-product-based operator evaluation can be implemented as matrix-matrix products.

The code is widely used in a broad range of applications. At the Department of Mechanics at KTH, which is a partner in CRESTA, the various research projects using Nek5000 include the study of the turbulent pipe flow, the flow along airplane wings, a jet in cross-flow and Lagrangian particle motion in complex geometries [2].

Nek5000 uses the Message Passing Interface (MPI) and domain decomposition for parallel computing on distributed memory supercomputers. The problem is spatially decomposed into many elements, which are distributed among the processing cores using MPI. The current version of Nek5000 supports conformal grids, in which element faces match each other, with uniform order of the spatial interpolations throughout the domain.

One of the main focuses within the CRESTA project is the implementation of Adaptive Mesh Refinement (AMR) algorithms to the Nek5000 code, which allows the possibility of increasing grid resolution in regions of the flow that are more dynamically active. Such a local refinement, either adaptive or by user intervention, is a desirable feature which will be crucial for the future scalability of the code, in particular for the simulation of large-scale problems involving turbulence. However, AMR constitutes a challenge as it can have a negative effect on the code scalability producing work-load imbalance between processors.

Within CRESTA the framework h-type refinement is going to be implemented. We have adopted a simple octree refinement algorithm and selected the p4est software library [4], which is designed to work in parallel and scale to hundreds of thousands of processor cores. Straightforward design and good scaling properties make this library suitable for CRESTA. The main task of p4est is the dynamical creation of the grid, and supplying Nek5000 with the full mesh information including element connectivity and global node numbering. Based on this, Nek5000 performs manipulation on simulation variables including data interpolation and element redistribution between processors. The main challenge in integrating p4est with Nek5000 is the different memory management and element ordering adopted by both codes.

Efficient solution of the pressure problem on a coarse grid in Nek5000 requires fine-tuning of the communication, and careful grid partitioning based on spectral bisection. This means that the grid data and the simulation variables for a given element reside on different processors and are differently ordered. The p4est-Nek5000 interface has to properly match grid and simulation data, and perform dynamic grid partitioning, which is essential for load balancing. The latter task is done with the ParMetis library, which is a parallel library for dynamical graph repartitioning. As an example, a mesh with 2D flow past circular cylinder on

four processes in Nek5000 is partitioned. The following Figure illustrates that ParMetis can obtain higher quality of partition compared with Nek5000 static partitioning.

OpenACC [3] enables existing HPC application codes to run on accelerators with minimal source-code changes. This is done using compiler directives (specially formatted comments) and API calls, with the compiler being responsible for generating optimized code, and the user guiding performance only where necessary.

In the CRESTA project, we ported NekBone, the skeleton version of Nek5000 to a parallel GPU-accelerated system. The initial profiling provided an assessment of the suitability of the code for use on a GPU system and also guided the optimization process. We then show the result with porting the full Nek5000 code to a single GPU accelerated system.

In Ref. [5], we presented a case study of porting NekBone to a parallel GPU-accelerated system. We used the original NekBone Fortran source code, enhanced by OpenACC directives. To port NekBone to the parallel GPU systems was relatively simple & primarily requiring the addition of a small number of extra lines of code. The naive implementation using OpenACC lead to little performance improvement in NekBone: on a single node, the original version of the code that was not using OpenACC reached 16 Gflops, and only 20 Gflops was obtained using the version with a naive OpenACC implementation. However, there was a significant improvement when we used an optimized version of NekBone & this lead to a performance of 43 GFlops on a single node.

In addition, we ported and optimized NekBone to a parallel GPU system, reaching a parallel efficiency of 68.7% on 16,384 GPUs of the Titan* XK7 supercomputer at the Oak Ridge National Laboratory. The scalability study is presented in Figure 1.

Currently, we are working on porting the full Nek5000 code to multi-GPU systems. In a first stage, we implemented the OpenACC version of the full Nek5000 code on a single GPU system. The so-called P_N-P_N algorithm has been employed in the OpenACC version of

Nek5000. In this implementation, the Conjugate Gradient linear solver with Jacobi pre-conditioner has been used.

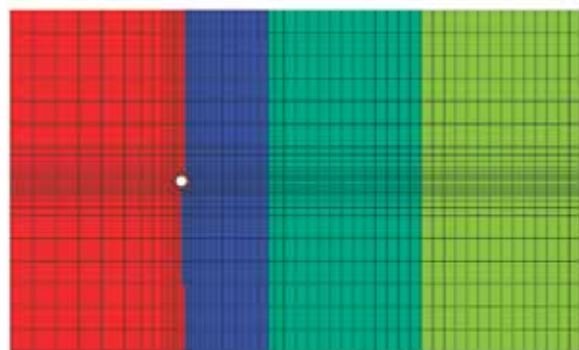
The following table shows the speed-up with OpenACC for a typical pipe-flow simulation with 400 elements on single GPU. In the table, E is the number of elements and N is the order of polynomial.

E = 400	CPU	GPU	Speed-up (GPU/CPU)
N = 10	41.66s / step	7.72s / step	5.4
N = 12	64.44s / step	9.95s / step	6.5

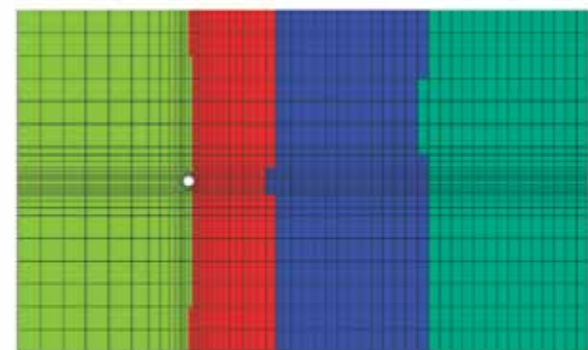
As visible in the table below, the resulting speed-up was significant. Within CRESTA, we focus on algorithmic improvements (AMR) and software challenges (using hybrid architectures) for exascale. The work described in this article addressed several challenges relating to adapting the Nek5000 code for use on GPU-accelerated systems (which are likely to become more and more prevalent in the future) and non-conformal meshes. We used OpenACC for the adaptations, rather than rewriting the code in a lower-level language (to avoid the disadvantages of multiple code versions).

We have integrated p4est and ParMetis with the current version of Nek5000 supporting conformal grids. Our first tests show ParMetis to be an efficient grid partitioner, however, there are significant differences with respect to the static, native nek5000 partitioner. To perform local refinement a Nek5000 version supporting non-conformal meshes is necessary. It is currently being developed by Paul Fischer (the main code developer).

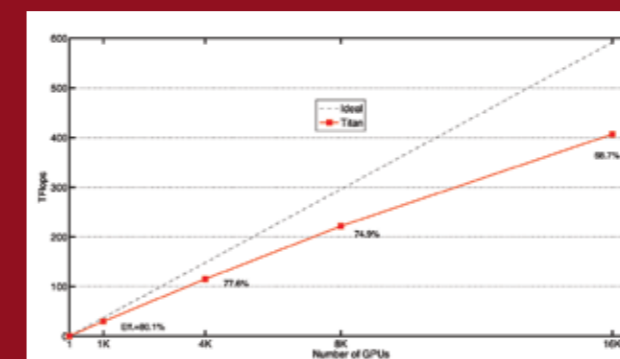
The NekBone code was successfully ported to a multi-GPU system using OpenACC compiler directives. The optimization aimed at reducing the data transfer between the host and the GPUs, increasing the speed of the computations on the GPUs and also enforcing parallel and sequential execution of the code. The tests of the version of NekBone optimized for multi-GPU systems demonstrated good scaling. The second part of this work consisted of implementing an OpenACC version of Nek5000 using the P_N-P_N algorithm.



ParMetis Graph partitioning



Nek5000 static partitioning



The weak scaling of NekBone with optimized subroutines on Titan

References

- [1] P. F. Fischer, J. W. Lottes, and S. G. Kerkemeier, The Nek5000 code, <http://nek5000.mcs.anl.gov>
- [2] The second nek5000 Users and Development Meeting, Zurich, Switzerland, 2012 <http://nek5000.mcs.anl.gov/index.php/Usermeeting2012>
- [3] OpenACC standard, <http://www.openacc-standard.org>
- [4] C. Burstedde, L. C. Wilcox, and O. Ghattas, p4est: Scalable Algorithms for Parallel Adaptive Mesh Refinement on Forests of Octrees. SIAM Journal on Scientific Computing, vol. 33, no. 3, 2011, p 1103-1133.
- [5] S. Markidis, J. Gong, M. Schliephake, E. Laure, A. Hart, D. Henty, K. Heisey, and P. F. Fischer, OpenACC Acceleration of Nek5000, Spectral Element Code, submitted to Advances in Engineering Software Journal.

* This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

OPENFOAM

MICHELE WEILAND, EPCC

OpenFOAM is an open-source toolbox for computational fluid dynamics (CFD). It provides generic tools to simulate complex physics for fluid flows involving chemical reactions, turbulence and/or heat transfer, in addition to solid dynamics and electromagnetism.

It contains a wide range of solvers that are required to simulate specific problems in engineering mechanics, in addition to the utilities required to perform pre- and post-processing tasks ranging from simple data manipulations to visualisation and mesh processing. Libraries are available to create toolboxes that are accessible to the solvers and utilities. The modular structure of OpenFOAM enables easy creation of solvers for multi-physics problems. One example is fluid-structure-interaction, where existing structural and fluid solvers or libraries can be coupled. This, of course, extends the complexity of the simulation, which leads to an increase in computational effort. The advantage of OpenFOAM compared to many other existing efforts in this field is that it enables easy and flexible implementation of different physical models, e.g. turbulence models, needed for numerical simulations, as well as their further modifications and development. This property makes OpenFOAM unique compared to other applications in this domain, and allows optimisations and extensions by a larger community. This allows, for example, being able to follow trends in turbulence investigation such as VLES (Very Large Eddy Simulation), and LS (Large Eddy Simulation), and combine them with improvements in other modules done in parallel by the community at large (e.g. within the PRACE context).

OpenFOAM is widely used in the automotive, aeronautical and heavy engineering industries, as well as being a staple resource for CFD modelling within the academic community. The renewable energy sector is also increasingly looking at CFD and OpenFOAM to model structures. For example, modelling the impact of both wind and tidal effects on the super-structure of an offshore wind turbine.

The Institute of Fluid Mechanics and Hydraulic Machinery (IHS) at the University of Stuttgart are using the capabilities of OpenFOAM in the field of numerical simulations of a complete hydraulic system in hydro power plants. Fluid flow through the hydraulic

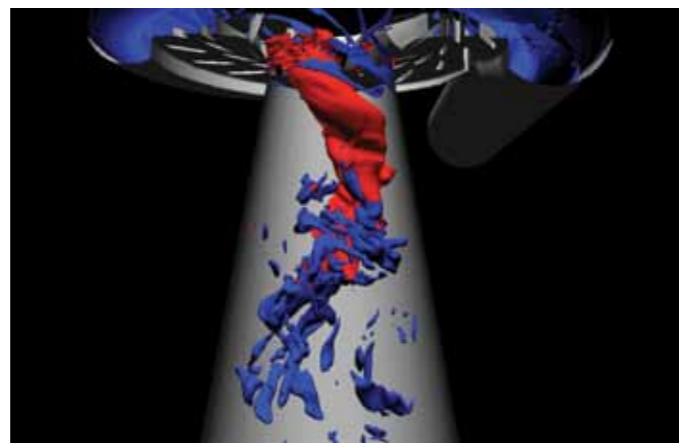


Figure 1 OpenFOAM simulation of turbulent flow inside a hydraulic system.

Image Sundance

system in water power plant is a very complex turbulent flow. The possibility to predict and understand turbulent flows is very important during the design process. Figure 1 shows a picture of a simulation of these turbulent flows using OpenFOAM.

Inside CRESTA, OpenFOAM is being applied and tested for the simulation of unsteady cavitating vortices in a hydraulic turbine. Because of a strong swirling flow, an accurate prediction requires sophisticated turbulence modelling. In this case a Large Eddy Simulation will be applied, which requires a very fine computational mesh. However, a highly accurate prediction of the swirling flow (especially the accurate prediction of the pressure drop in the vortex center) combined with a sophisticated cavitation model is necessary to predict accurately the right behaviour. Because of the unsteady flow and because of applied rotor stator interactions together with the very fine mesh and the sophisticated models, the computational effort is quite enormous.

CRESTA is working on enabling these large-scale simulations on next-generations HPC systems by specifically addressing the following points:

- **IO:** an application like OpenFOAM needs to output a lot of results, both during and at the end of a simulation. Creating and writing to very large numbers of files leads to expected bottlenecks for Exascale simulation. CRESTA is therefore exploring alternative implementations of IO for OpenFOAM, which will focus offloading this task to specialist cores and overlapping it will computation in an asynchronous fashion.
- **Numerical solvers:** CRESTA is developing new numerical solvers and is testing their performance in terms of accuracy, scalability, time-to-solution and fault tolerance. The problems they are being tested on are extracted from OpenFOAM, thus enabling a direct comparison with the solvers that are already part of the application.
- **Performance analysis:** OpenFOAM is a very complex code, and some of its implementation characteristics (such as the heavy use of C++ templates) make it a challenge for many performance analysis tools, especially at a large scale. CRESTA uses OpenFOAM as an example of a demanding application to improve on the state-of-the-art in performance analysis techniques.

CRESTA AT SC13

KATIE URQUHART, EPCC

CRESTA is looking forward to once again being well represented at this year's SC13 which takes place from 17th - 22nd November in Denver, Colorado.

Along with the DEEP and Mont-Blanc projects, we are running a Birds of a Feather session entitled 'Building on the European Exascale Approach. All three projects are now well established, making a comparison of Europe's exascale approaches with that of other countries and continents particularly timely. In this session we will debate these approaches and look to build cross continent communities. The BOF runs on Tuesday 19th November from 12.15pm - 1.15pm.

Following the success of the workshop from last year, CRESTA has a workshop at SC13 on 'Exascale MPI.' The aim of workshop is to bring together developers and researchers to present and discuss innovative algorithms and concepts in Message Passing programming models, in particular related to MPI. This is being held in collaboration with the EPIGRAM project (a newly funded FP7 Exascale project led by KTH). The workshop runs on Friday 22nd November from 9.00am - 1.00pm.

CRESTA partners Allinea Software and Technischen Universität Dresden are hosting a tutorial on Allinea's flagship debugger DDT and TUD's MUST. The tutorial, 'Debugging MPI and Hybrid/Heterogeneous Applications at Scale' will take place on Sunday 17th November 8.30am - 5.00pm. Allinea will also be presenting a talk at a workshop on 'Extreme-Scale programming tools' on the 18th November 9:00am - 5:30pm.

CRESTA partner Cray is hosting a tutorial on 'OpenACC: Productive, Portable Performance on Hybrid Systems Using High-Level Compilers and Tools' which also runs on Sunday 17th November from 8.30am - 5.00pm.

A highlight for CRESTA at SC13 is our joint booth with DEEP and Mont-Blanc, entitled 'European Exascale Projects' (booth number 3741). The booth will display highlights from all three projects and we will have a range of new flyers and information on the project available. A range of CRESTA project partners will staff the booth at all times and we will be providing a program that showcases different aspects of the projects in turn. Please do come along and visit us at booth number 3741 where we will be happy to chat with you more about the project.

Finally, a number of our partners are exhibiting at this year's event, these include: Allinea Software (1719), Cray (1313), CSC (1913) and EPCC (3932). All are keen to welcome you at their booths so please do drop by.

We look forward to seeing you in Denver!

OTHER EVENTS

PGAS@EPCC

NICK JOHNSON, EPCC

The 7th International Conference on PGAS Programming Models was held on the 3rd and 4th of October in Edinburgh, hosted by CRESTA partner EPCC and sponsored by NAIS, Edinburgh Global and The University of Edinburgh. This was the first time PGAS has been held outside of the US and was a great success with over 60 delegates from industry and academia coming from North America, Europe and Asia.

Both days were packed with high quality presentations on developments in hardware support for PGAS, programming models and standardisation efforts and applications under development. There were 27 talks of which roughly half were in the research category for mature work where authors were reporting substantive results, and the remainder in the hot category where authors were able to seek feedback and proffer early results on work in active development.

In addition, there were two preceding days of tutorials at EPCC covering UPC, CAF and mixing MPI with PGAS models with many delegates coming for the whole week.

For those interested in the research presented, a full proceedings is available on the conference website:

www.pgas2013.org.uk

EASC2013

ADRIAN JACKSON, EPCC

The first Exascale Applications and Software Conference (EASC2013) was held in Edinburgh at the beginning of April 2013. Conceived to provide a forum for the discussion and presentation of exascale issues and science, the conference was hosted by EPCC and supported by the CRESTA project (along with the NAIS and Nu-FuSE projects in the UK, both of which also are tackling exascale issues for scientific applications), with the goal of bring together all of the stakeholders involved in solving the software challenges of the exascale - from application developers, through numerical library experts, programming model developers and integrators, to tools designers.

The conference was a great success, with over 130 participants from around the world attending two and a half days of a high quality technical program. This included keynote talks from Satoshi Matsuoka (Tokyo Institute of Technology), Vladimir Voevodin (Research Computing Center, Moscow State University), Bill Tang (Princeton Plasma Physics Laboratory), George Mozdzynski (ECMWF), Peter Coveney (Centre for Computational Science at University College London),

and Jack Dongarra (Electrical Engineering and Computer Science Department, University of Tennessee).

There was also a panel session where some of the invited speakers and other well known researchers in the high performance computing field discussed the challenges of exascale computing for users and developers.

All of the technical and keynote presentations from the

conference are available online at <http://www.easc2013.org.uk/> programme and a selection of the best papers will shortly be published in a special edition of the Advances in Engineering Software Journal.

Building on the success of the first EASC conference, we are delighted to report that EASC2014 will take place from 2nd - 5th April 2014 at CRESTA partner KTH, in Stockholm. We hope to see you there!

CRESTA ON SCREEN

KATIE URQUHART, EPCC



CRESTA has recently produced an exciting series of short videos which explain the innovative work being undertaken within the project.

Our key challenge for making the CRESTA films was to explain highly complex technology and applications to policy makers, industrialists and the general public and, through good storytelling, enable the CRESTA mission to relate to people's everyday concerns in a memorable way.

In order to meet this challenge, each of our five films takes a CRESTA application and, through a series of short interviews, explains 'real-life' uses for, and the importance of, the application, all in an engaging and easy to understand manner. The 'interview-style' concept that we employed throughout each of the films offers energy, interaction between partners and co-designers, and sets a tone that we think reflects a successful collaborative project talking about itself as a community.

We have produced the following five films:

- An Introduction to Exascale
- An Introduction to HemeLB
- An Introduction to IFS
- An Introduction to GROMACS
- Modelling for Large Engineering Projects

All of the films are available to view from our website:

www.cresta-project.eu

We thoroughly enjoyed making these films and thank euConnect for bringing them to life. We hope that you enjoy watching them and that you find them both informative and entertaining!

CREST 