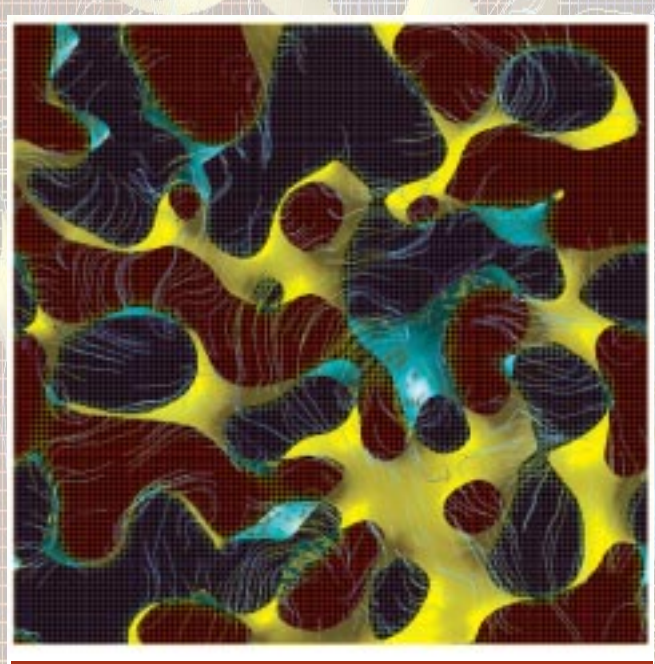
ISSUE 4, SUMMER 2002

UKHEC

The newsletter of the UK high-end computing initiative



The interface between two fluids undergoing spinodal decomposition in the interial demixing regime.

Contents

Editorial

Lorna Smith, EPCC

Welcome to the latest UKHEC newsletter.

One of the main aims of the UKHEC collaboration is to disseminate information to you, the high-end computing community. The past few months have seen a fair amount of activity in this area, with all centres running various workshops and seminars. One of my particular highlights has to be the 2nd UKHEC annual seminar, which was very well attended. I would like to thank all of you who made the effort to visit the new e-Science centre in Edinburgh, I hope you found the event as interesting as I did. Dates for the 3rd annual seminar have already been decided, so please mark the 10th and 11th December 2002 in your diaries.

I would also like to draw your attention to the article on Supercomputing, the foremost international conference on high performance networking and computing. Held in Denver, all three centres participating with booths, technical papers, tutorials and access grid events.

I hope you enjoy this issue.

Lorna Smith

CLRC CENTRAL LABORATORY OF THE RESEARCH COUNCILS

epcc

MRCCS

Next generation HPC systems and the Grid

Lorna Smith, EPCC

The Second Annual Seminar took place in Edinburgh, during September 2001. The meeting, which was sponsored by Sun Microsystems, was well attended with over 110 registrations. The event was held within the newly formed e-Science Centre, which offered excellent conference facilities.

The seminar was spread over two days. The first day focussed on programming next generation High-End Systems, a topic of particular interest in the UK with the current HPC(X) procurement underway. In addition to talks from the UKHEC centres, there were a number of excellent talks from invited speakers. Mike Vildibill, the deputy director from San Diego Supercomputer Centre, gave an interesting talk on the TeraGrid project, a recently funded venture between San Diego and the National Centre for Supercomputing Applications to create a Distributed Terascale Facility (DTF). The talk was particularly relevant to current UK activities with a focus on the design decisions and anticipated challenges associated with deploying a national Grid infrastructure.

Interesting talks were also provided by Hugh Pilcher-Clayton from the EPSRC who brought us up to date on the progress of the HPC(X) procurement and from Bryan Richards and Mark Woodgate of Glasgow University, who focussed on their considerable practical experience of programming Beowulf clusters to tackle real engineering problems in aerodynamics.

The emphasis of the second day was slightly different, and focussed on Exploiting the Computational Grid. Tony Hey, the Director of the UK e-Science Core Programme, started the day by considering 'e-Science, e-Business and the Grid'. This talk was followed by two interesting presentations, one from Steven Newhouse of Imperial College on resource exploiting within the Grid, and the other from Rob Baxter of EPCC who presented results of the EPCC ePortal, a user-driven Grid portal for large-scale computational e-science.

The afternoon session began with a talk from Jarek Nabrzyski, from Poznan Supercomputing and Networking Center. His talk focussed very much on the users' view of the Grid, mapping user requirements to existing Grid services and highlighting any deficiencies. His results were based on the Grid User Requirements questionnaire produced under the ENACTS project (see below for web page).

The day finished with talks from Jon MacLaren of Manchester Computing and Rob Allan of Daresbury Laboratory. Jon focussed on the UK Grid and on EUROGRID, a project aimed at linking different machines into a European-wide HPC-GRID. Rob meanwhile reviewed the UKHEC activities in Grid computing with a particular focus on web portals and expert systems.

Mike Vildibill explaining the TeraGrid project during the second UKHEC Annual Seminar.

Full details of the event, including copies of the speakers presentations, can be found at:
http://www.ukhec.ac.uk/events/annual2001/

Grid User Requirements questionnaire results:
http://www.epcc.ed.ac.uk/enacts/activities/gridservice/

How long is a piece of string?

Kostas Kavoussanakis, EPCC

A UKHEC report on software estimation has recently been produced by EPCC. While EPCC has not had traumatic experiences with software projects, this is not the case for the industry in general.

Many projects are haunted by over-optimistic functionality targets, unrealistic time estimates and impossibly mean budgets, and this is certainly not restricted to the software industry. The 'Cost, Quality, Time' triangle is applicable in all development contexts. The effects of this behaviour are overwhelming. Under-staffing, short-changed quality assurance methods and minimal schedules are followed by staff burnout, quality compromises and overall loss of credibility. Often the culmination is for the project to be cancelled but not before large sums of money have been spent.

The question is why we allow such tragedies to happen? Here we focus on a way to prevent them: estimation.

What is estimation?

Estimation is the complex process of generating meaningful measures for the effort, schedule and functionality of a (software) project. This is a fundamentally flawed process, because the estimates are most useful before the work starts, which is also when they are most inaccurate. Studies show that an estimate at the feasibility study stage of a software project may be four times off the final figures. Yet we base our business on this inaccurate prediction.

There has been a lot of work in the estimation area, in particular with a view to develop metrics for the objective measurement of the quantities that concern us. The results have been encouraging; however, the focus has been on MIS and recently on real-time systems. Academic research projects involving complex computations are very unpredictable. A relatively new method (Full Function Points) explicitly excludes them from its applicability domain (although it defines parameters for pertinent tuning).

The road to success

Despite the shortcomings, estimation is essential. No one will trust you with their money unless they are given a breakdown of the quality, time and cost requirements. The way to minimise suffering from the inherent inaccuracies of early estimation is to accept the fact that it is flawed and seek contingencies. Estimation is only a small part of the software process and cannot be seen in isolation. Even if the metrics and tools do not work particularly well for the academic world, they are still useful, mainly because most of them provide tuning mechanisms for the user to adapt them to their circumstances. It is very important to consider what these circumstances are. The tools in any case are invaluable if only for the reason that they encourage careful thinking about all the stages of the project.

It is good practice to identify a development model, preferably an iterative one. A good example is the case of a fixed effort, fixed cost project. In this case, it is recommended to select a model that develops functionality gradually; this needs to be coupled with a prioritised list of features implemented in descending order of importance. You will need to work with your client so as to agree these priorities.

It is essential to keep track of the risks involved in a project. In addition to its conventional use, the risk-list can be utilised when the time comes to estimate stages of the project. After breaking the project down into workpackages, you can assess the possible impact of the risks on each specific component. This leads to the so-called 'estimation in range', whereby a task 'will take [6 +2 -3] months, according to the realisation and effect of risks 2 and 8 and the extent of the benefit from the use of methodology X'. In this context, 6 +2 -3 cannot be aggregated to 5.

There is a lot to be gained from introducing the developers to the estimation process, as this gives a feeling of ownership and affection to the project. After all, it is they who will have to do the work. And after you have done the sums, make sure that not much work is expected in the week 25/12-1/1 and plan for staff holidays and sickness.

This UKHEC report is available from:
http://www.ukhec.ac.uk/publications/reports/estimation.pdf

The EPCC web-based prototype ePortal.

Investigating the use of XML in Java application codes

Lindsay Pottage, EPCC

XML (Extensible Markup Language) and Java constitute the core of Web services today. Java provides the tools to develop secure, portable and platform-independent Web-based applications, and XML their means of communication via structured, platform-independent data representation and exchange.

XML is a restricted form of SGML (the Standard Generalized Markup Language) specially designed for Web applications. SGML, in general terms, is a system which enables documents to describe their own grammar and structural relationships via a set of tags. However, the full SGML standard is very complex and contains many infrequently used features.

XML was developed to address the limitations of HTML (Hypertext Markup Language), the now familiar SGML-based language used for storage and transmission of documents over the web. HTML is a specific application of SGML, which defines simple hypertext and multimedia supportive documents of fixed type via a hardwired SGML-conforming tag set. However, due to its simplicity, HTML cannot offer functionality such as extensibility, structure and validation, all of which are required by Web applications today. Subsequently, XML was specifically devised to retain these features already provided by SGML via a simplified language designed for straightforward use over the Internet.

It is worth noting that any programming language can output data from an XML document; however Java offers the most compatible environment to XML in terms of platform-neutrality, object-oriented structure and the industry-wide adoption of such technologies as Enterprise JavaBeans, RMI, JDBC and servlets for distributed network and Web-enabled applications.

In this UKHEC technical report we therefore focus on XML and Java and provide a practical overview of how to use XML in user applications. The report describes how XML and Java work together using, as a case study, software recently developed at EPCC for a Web-based prototype ePortal to HPC applications.

See:
http://www.ukhec.ac.uk/publications/reports/xmlreport.pdf

SC2001

Denver, Colorado
10–16 November

David Henty, EPCC

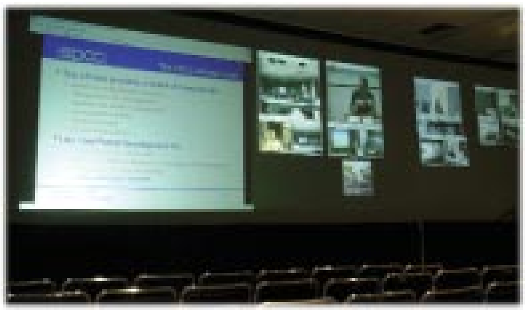
Supercomputing is the foremost international conference on high-performance networking and computing, and the UKHEC sites were involved in a whole range of activities. We all had Research Booths in the main exhibition hall, grouped within the new 'European Village'.

In addition to contributing research booths, members of UKHEC also presented research work as part of the Technical Program, ran a workshop over Access Grid as part of SC Global and held a Tutorial session on Java for HPC.

I'd first like to try and give an overall impression of the entire week-long event in a paragraph. It's clear that the surge of interest in the Grid has had a major effect, and the majority of stands in the exhibition hall mentioned the Grid in one way or another. Researchers were keen to showcase their Grid projects, software vendors described how their products would operate in a Grid context and hardware vendors promoted their own technologies as the key Grid platforms. The impact on the technical programme was not as dramatic, although almost a quarter of the technical sessions did have 'Grid' somewhere in the title. However, the focus was still very much on the Computational Grid with not much in the very important area of Data Grids. It will be interesting to see if this changes in the coming year.

I was also pleased to see that SC2001 still had space for good old-fashioned parallel supercomputers! Compaq, IBM, Intel, SGI and Sun were all demonstrating new hardware, and the Moore's law speed-up curve remarkably still shows little sign of tailing off.

Overall, SC2001 was extremely enjoyable and I hope that the UK HPC community can maintain the same level of participation in next year's meeting in Baltimore.



SC Global *David Henty, EPCC*

John Brooke of MRCCS coordinated a session on 'Solar Terrestrial Physics' which was a global, interactive event using Access Grid technology to link Denver with other sites including Manchester and two US national labs. Sitting in the US we were able to see Rob Baxter of EPCC give a talk on 'The EPCC ePortal: Grid Support for Solar Magnetohydrodynamics' live from the UK, work which lead to a recent UKHEC report on XML. Joanna Leng also spoke from Manchester on Visualisation of solar flares, and there were additional contributions from the Arctic Region Supercomputing Center.

Although it is clearly still an evolving technology, SC Global illustrated the potential of the Access Grid to enable truly distributed workshops with a level of interactivity simply not possible with traditional video-conferencing. As the number of UK sites continues to grow, I am sure that it will become a standard way of opening up meetings beyond the local venue and out to a much wider networked audience.

Java for high performance computing

Lorna Smith, EPCC

As part of SC2001, EPCC presented a tutorial on Java for high performance computing (HPC). The tutorial involved lectures and a number of practical coding sessions which reinforced the concepts described in the lectures. The feedback was generally very positive, with the practical sessions being particularly successful. We are grateful to Sun Microsystems for loaning SunRay terminals for the duration of the day.

The first lecture focussed on the potential benefits of Java as a language for HPC, especially in the context of the Computational Grid. For example, Java offers a high level of platform independence. This is important in an area where the lifetime of application codes exceeds that of most machines. In addition, the object-oriented nature of Java facilitates re-use and reduces development time.

There are however a number of issues surrounding the use of Java for HPC, principally performance, numerics and parallelism. EPCC is leading the Benchmarking initiative of the Java Grande Forum, which is specifically concerned with performance. The remainder of the tutorial focussed on this work, examining performance issues relevant to HPC applications. We considered benchmarks for evaluating different Java environments, for inter-language comparisons and for testing the performance and scalability of different Java parallel models.

The aim was to demonstrate that performance no longer prohibits the use of Java as a base language for HPC.

For those that are interested, this tutorial will form part of the MSc module 'Object-oriented Programming for HPC'. Further details can be found at: http://www.epcc.ed.ac.uk/msc/

In addition, a UKHEC technology report on Java for HPC provides further details on the subject, see: http://www.ukhec.ac.uk/publications/tw/hpcjava.pdf

Finally further information on SC2001 can be found at: http://www.sc2001.org

TECHNICAL REPORT: Web and Grid services

R.J. Allan and D.J. Hanlon, Daresbury Laboratory

Emerging standards such as XML and SOAP are enabling a new generation of 'Web Services' that allow systems to be self-describing and to make remote procedure calls to other systems over the Internet. A business example is for a corporate inventory management system to publish a service that allows a customer system to check real-time inventory levels.

In the wider sense, services are what we define in terms of logical (as opposed to physical) integrated system components. There could be more than one component for each service, so a service could be an 'aggregation' of components. We consider a portal to be both a user interface and a means of managing aggregated components.

The recent announcement of the Open Grid Services Architecture gives us insight into how the Grid API may be accessed as a set of persistent Web services or time-limited Grid services. We have tried to include a description of how Grid Services might help with development of an e-Science environment.

This report provides an overview of Web and Grid Services.

Technical report available from: http://esc.dl.ac.uk/TechReports/WebServices/webServices_doc.pdf

REVIEW

The Viz 2001 conference

Joanna Leng, The University of Manchester

The Viz 2001 series of conferences is sponsored by the IEEE Computer Society Technical Committee on Visualisation and Graphics, the third biggest IEEE technical committee. It has run for approximately 10 years and is the biggest Visualisation conference of the year. It started as a conference aimed at HPC service providers with many hands-on tutorials but has progressively become more oriented towards academic research.

It is now a six-day conference. The first three days are for tutorials of which the last two days also have symposia running in parallel. There is a well established two-day symposium on Information Visualisation and this year the second symposium was on Parallel and Large Data Visualisation. The final three days of the conference was a three-track event on Scientific Visualisation, one each for techniques, algorithms and applications. There were also case studies, panels and an exhibition.

Parallel Techniques

Graphics hardware is currently the fastest developing hardware compared to processors, hard disc or memory. Until recently the bottleneck in the visualisation pipeline has been in the rendering but the improvement in the graphics hardware has pushed the bottleneck higher up the visualisation pipeline, for example into the isosurfacer.

Strategies for parallelising particular visualisation modules or for streaming the data through the pipeline have recently been implemented in many visualisation systems, so that the machine's cache is never completely used up. The one day tutorial 'Large Scale Data Visualisation and Rendering' focussed on this topic. There was a strong contingent of VTK users presenting the tutorial and the bias was toward the solution of this problem, but OpenDX and Chromium were also discussed.

Surface Rendering and Multi-resolution Techniques

Although graphics hardware will improve, the size of the data sets will also increase so that at any time it is likely that there are data sets that are too large for current hardware. There are several types of techniques that aim to improve the speed of rendering.

A big topic this year was point rendering which is where a surface is represented as points instead of the triangular mesh commonly used in graphics.

A point is relatively 'light weight' and can be defined by its 3D coordinates and its colour. A triangular mesh has coordinates for each apex defined in 3D space and an associated colour but the mesh has extra information – the connectivity between each apex – and, when rendered, the colour and light must be interpolated across the surface of each triangle. With point rendering if there are enough points, when the camera is far away there will be at least one point for each pixel of the display and the resolution will be high enough for the user to perceive it as a surface. However, if there are too few points or if the camera is close, gaps will be seen between the points. Some algorithms have been developed to fill in gaps between points for close positioning of the camera. There was some feeling that this technique would be good for computer games and may soon be incorporated in graphics cards.

Level of detail techniques may also be useful for large data sets. If objects defined as meshes are in a visualisation scene then the level of detail of the mesh can be different depending on the distance from the camera. If the mesh that represents a bunny, for example, is far from the camera it may be possible to represent it with 100 triangles but if it is near the camera it may need many more.

There are many techniques for reducing the size of a mesh while keeping the overall topology of the object. The problem with these techniques is that they are often data dependant and can not be implemented in hardware. Mesh decimation algorithms are compute intensive and so are often implemented in parallel to make the application more interactive. However they are best not used on the fly but used to reduce the complexity of a data set before it is read into a visualisation system.

Art and Visualisation

There was a panel entitled 'Realism, Expressionism, and Abstraction: Applying Art Techniques to Visualisation' chaired by Theresa-Marie Rhyne. The panelists showed some interesting work and explained how they had taken techniques from art. Each panelist has worked using the principals of art and human perception to make the comprehension of their visualisation images easier, more 'intuitive'.

The panel ended with a rather heated discussion. I think some of the audience mistakenly thought that the

Consolidated Globus evaluations

R.J. Allan, Rob Baxter, D.R.S. Boyd, John Brooke, Mike Daw, T. Folkes, Jon Gibson, C. Greenough, D. Hanlon, David Henty, W.T. Hewitt, R.P. Middleton, Elson Mourao, Mark Parsons, Jon MacLaren, Stephen Pickles, Graham Riley, R.A. Sansum.

We have, over the past 18 months or so, installed and evaluated a range of middleware and other software to enable development of Grid-enabled services and interfaces for e-Science applications. These have been applied in a number of projects, for instance in the HPCGrid Portal, JiniGrid, EuroGrid, GridPP and European Data Grid (EDG).

In this report ('Globus and Associated Grid Middleware: Consolidated Evaluation Report from UKHEC Sites') we comment on aspects of the installation and maintenance of the middleware currently available and its functionality. We summarise our experiences of using the software in a wide range of projects and indicate how it can be used to further facilitate UK collaborative research based on e-Science technology. We make reference to related middleware projects in the USA, Europe and Australasia in which we are partners. We finally suggest some additional specific issues which must be addressed in the UK to improve uptake and extend functionality, perhaps tackled as part of the workplan of the Research Council funded 'pilot' projects.

Further comments are made on more general topics which we believe to be of particular importance in the deployment of robust and pervasive computational Grid solutions on a national scale. These include security, certificate authorities (authentication), registration (and authorisation), validation and accounting.

Appendices in the original reports from CLRC, Edinburgh and Manchester contained detailed descriptions of the evaluation projects undertaken and reported on the individual software components. They also included contributions from our collaborators who are involved in complementary e-Science and Grid projects. These detailed accounts are time-bound and will not be repeated here. We have however complemented the original material by adding comments on the current situation regarding Globus v2.0 beta. Many of these comments were provided by the Globus Team in a detailed response to the original CLRC report. We also update the status of ongoing projects of the three UKHEC partners.

The report can be downloaded from:
http://www.ukhec.ac.uk/publications/reports/consolidate.pdf

Viz 2001 conference

continued

panel were advocating an 'artistic' attitude toward visualisation where feelings were being expressed, rather than scientific data. The panel saw art as a very structured discipline and thought concepts in composition and image detail should be used to produce better understanding. In any case it is true that the evaluation of a visualisation and its metaphors is not scientific and is done by user evaluation – a method comparable to reviews or criticism in art.

Virtual Reality

This visualisation community tend to think of VR/VE as large display systems with stereo imaging. I would define VR as systems that try to be immersive and use feedback controlled interfaces to achieve this. The interfaces could be visual (head tracking), haptic or sound. In my opinion VR is not when for example haptic feedback is added to a visualisation because the visual channel is overloaded.

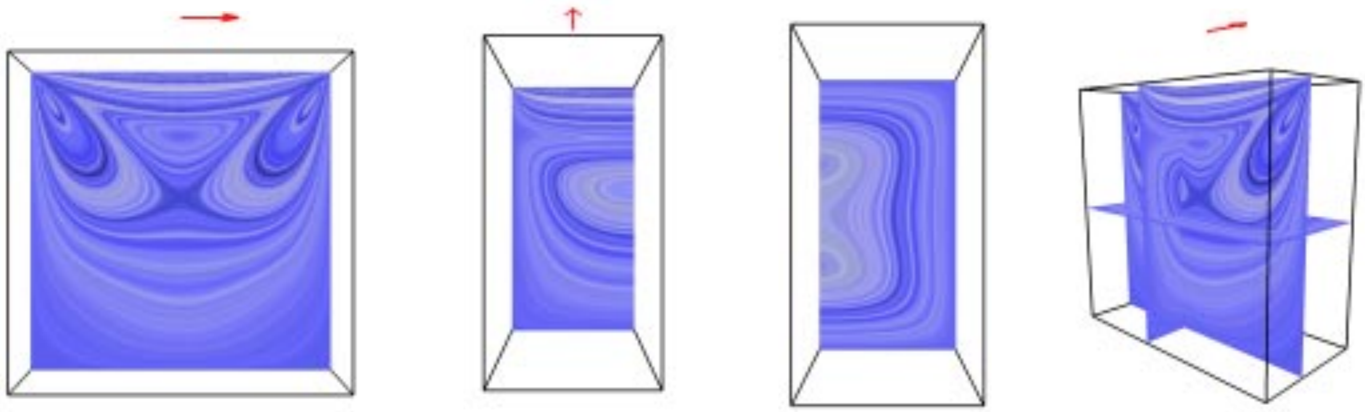
This year there were four case study papers in only one VR session and although all four were strong papers only two of these would fit my definition of VR.

Information Visualisation

Generally the scientific visualisation community is separate to that of the information visualisation community. However there is some overlap.

Information visualisation tends to work with data bases and produce images related to graphs (often trees or circular displays) although it has been used to produce systems to organise email and debug software.

A fuller review of this conference is given on the UKHEC website where references are given to specific papers, other websites and useful visualization resources (http://www.ukhec.ac.uk/publications/trips/viz2001.html).



Using Green to solve the Navier Stokes equations in parallel

Joanna Leng, Lee Margetts, Mike Pettipher (MRCCS, Manchester Computing), Ian Smith (Manchester School of Engineering)

The Lid-driven Cavity

To demonstrate the performance of a parallel Navier Stokes solver developed at the University of Manchester, a lid-driven cavity problem was solved. The results were visualized as a part of a UKHEC case study and shown in an immersive visualization environment at the inauguration of Green, the CSAR flagship 512 processor Origin3800, in April 2002.

The lid-driven cavity problem is a well documented test case for computational fluid dynamics (CFD) algorithms. A cubic cavity contains a fluid that is initially at rest. The top surface of the cavity or 'lid' is driven at a constant velocity. A steady state solution is then sought for the motion of the fluid inside the cavity.

The domain was subdivided into a quarter of a million finite elements, giving rise to 1 million grid points where values for the pressure and velocity field were to be calculated. With approximately 4 unknowns at each grid point, the computational task was to solve a system of 4 million non-linear simultaneous equations.

Problem Size	256 000	20 node brick elements
	1 000 000	Nodes or grid points
	4 000 000	Simultaneous equations

A Range of Solution Methods

There are various ways of solving CFD problems that typically involve numerical techniques such as finite differences or finite elements. Various formulations are in use that simplify the Navier Stokes equations by taking mathematical short cuts. For simple test problems, these methods produce similar answers with reasonable computation times. However, for complicated geometries, these simplified methods may not give the correct answer or give any answer at all.

The closest we can get to the correct solution is to solve the full system of simultaneous equations with no simplifications. This is often referred to as direct numerical simulation or DNS. Being computationally very expensive, DNS is not typically used for everyday CFD problems. However, solving large complicated problems using DNS on supercomputers such as Green benefits the developers of simplified industrial algorithms by providing accurate solutions for validation work.

Background to the Program Development

Commercial finite element packages would have great difficulty solving this size of Navier Stokes problem by DNS simply because of memory constraints. Typically a global stiffness matrix is assembled for the whole domain and the equations are solved using some form of gaussian elimination. Solving a problem of the size described here would involve much paging to disc, resulting in very poor performance. One should bear in mind that the parallel program used in this report is optimised to keep much of the data needed for the computation in cache.

The original serial program was written by Ian Smith. In order to avoid the memory limitations of commercial

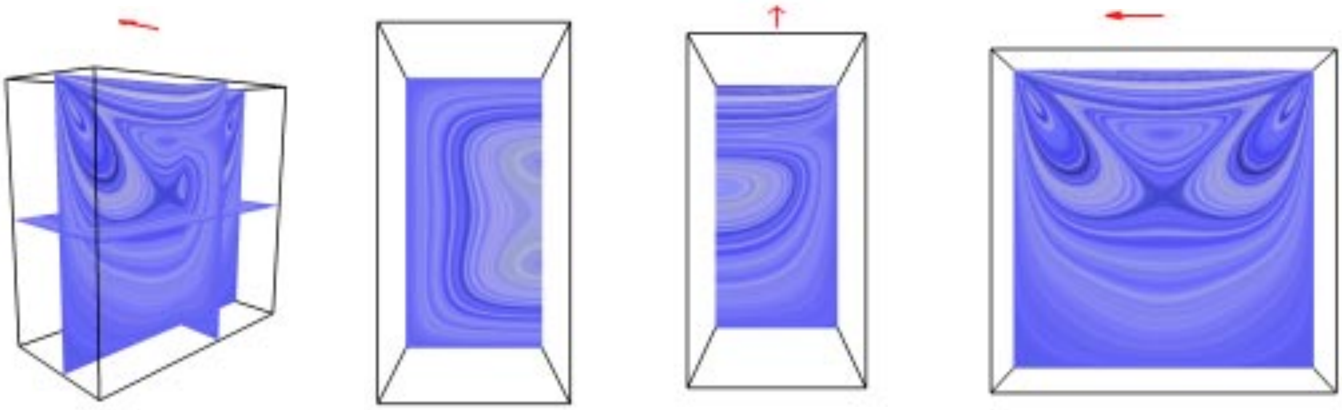


Table 1: Performance data versus Reynolds number.

Problem Number	Reynolds	Parallel Time Minutes	Serial Time Days	%Peak Performance	Gflops
Half Cubic Cavity	10	20	2-3	29	59
	100	47	8-9	29	59
	1000	180	1 month	29	59

approaches, the global stiffness matrix is never constructed; the problem is solved element by element using the iterative solver BiCGStab(l). The program was parallelised by Lee Margetts using MPI subroutines originally developed by Mike Pettipher at Manchester Computing. Recently, Lee Margetts joined Mike Pettipher to parallelise a whole suite of engineering programs developed by Ian Smith's research group.

The parallel versions of the driver programs are virtually identical to the serial ones with all the parallel libraries hidden from the end user. Hence they should have mass appeal and are intended for use by post-graduates or researchers in engineering research projects. The researchers need have little interest in parallel computing, apart from enjoying the benefit of fast computation times. The programs can also be used as teaching material alongside the serial programs in undergraduate modules of finite element analysis.

Program Performance

Table 1 shows the performance data for the problem described, analysed using different values of Reynolds Number. The parallel time is for 256 processors, and the serial time is estimated from the time per iteration. It should be noted not only that the parallel version runs 256 times faster than the serial, but also that the actual performance relative to the peak performance of the processors is also very good.

The Reynolds Number is a dimensionless parameter that characterises the flow. In simple terms, one can imagine a low Reynolds flow being laminar, like treacle and a very high Reynolds flow being turbulent, like smoke rising from a cigarette. As the Reynolds number increases, the computational cost increases and at a certain point a solution is unobtainable. Increasing the resolution of the analysis (number of grid points) enables the computation to proceed to a higher Reynolds number. To resolve fully turbulent flows using DNS, it has been suggested that around 1 billion computational grid points would be needed. We're a long way off at 1 million!

Visualization

The speed up of the computation has meant that more data at higher resolution is being produced than ever before and now processing this data is the bottleneck of the computation. All features of the data, both expected and unexpected need to be found and analysed painlessly. Doing this not only requires the use of high-end interactive visualization equipment but also the application of appropriate visualization techniques. Producing an output compatible with AVS/Express, an interactive visualization system, is in the process of being developed as a UKHEC case study by Joanna Leng.

This work was funded by EPSRC Award 98317397

JAVA FOR HPC, A CASE STUDY: Porting LUDWIG to Java and OpenMP

J.C. Desplat, EPCC

With the emergence of Grid computing as the hottest topic in high-performance computing, the importance of the true binary portability offered by Java has strengthened the case for reassessing the suitability of this language for Grande application codes. However, a recent study undertaken amongst major European user groups in computational science and engineering has shown that Java was still considered by most to be highly unsuitable for HPC [1]. When asked if they would consider porting their existing codes to Java or choose this language to develop new codes, only 15% of the respondents said that they might. Reasons given by the others not to do so included its (perceived) poor performance (41%), and the large overhead involved in developing/porting codes to this language (38%).

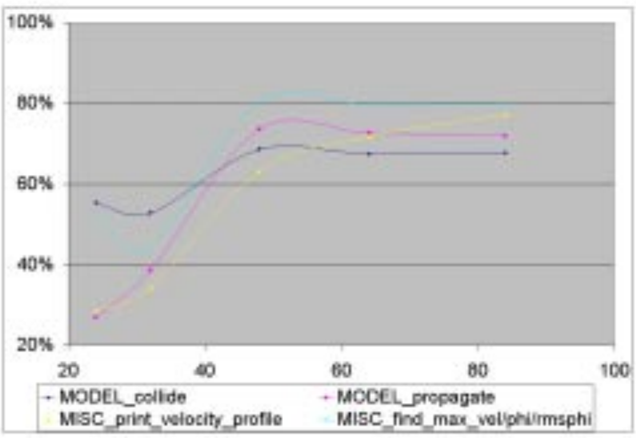
It is within this context that the author undertook to investigate the suitability of Java for a Grand Challenge application code he had previously developed for Prof. M.E. Cates' Condensed Matter group at the University of Edinburgh. This code, known as 'LUDWIG', is a generalised Lattice-Boltzmann code for the simulation of the hydrodynamics of complex fluids in 3-D. It comprises over 25,000 lines of C and can use either MPI or OpenMP to deliver optimal performance even for load imbalanced problems. The two main routines (in which 90% of the runtime is spent) consist of a series of basic arithmetic operations embedded within a large triple loop (MODEL_collide), and series of memory-to-memory copies with different strides (MODEL_propagate). Further information about this code and its applications can be found in the literature (e.g., see [2]).

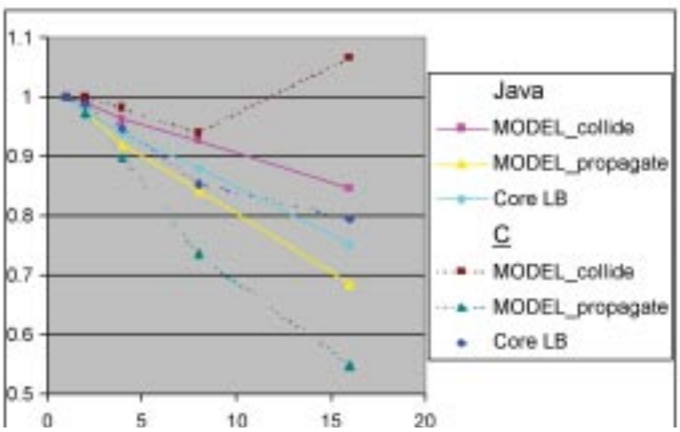
At the start of this project, Rúben García-Hernández successfully ported a cut down version of LUDWIG (still over 5,000 lines long) within only two weeks. This quick conversion was made possible by exploiting the syntactical similarities between C and Java to maximise the use of 'cut & paste' operations.

All benchmarks presented in this article were carried out on EPCC's Sun Fire 6800 with Sun Workshop 6 C5.3, and Sun JDK1.4b3-client. As shown in Figure 1, the relative performance of Java vs. C for serial runs significantly varies with the system size. Whilst the performance of Java appears relatively poor for small system sizes, it progressively increases to reach a stable relative performance of 65% to 80% of that of C depending on the function. The poor performance for systems below 32^3 (for which most of the data fit in cache) is thought to be caused by the larger number of instructions generated by Java to perform any given tasks. This increased number of instruction loads obviously becomes more noticeable in situations where the number of memory access decreases. Typically, however, codes do not use these low system sizes.

The Java implementation was then parallelised using the JOMP library (OpenMP for Java) developed by Dr Mark Bull of EPCC [3]. A direct comparison of the C and Java OpenMP implementations (see Figure 2) reveals that both codes exhibit similar scaling properties, with typical parallel efficiencies of 80% up to 16 processors. The only noticeable difference stems from the failure of

Figure 1 (below left): Java vs. C for various system sizes
Figure 2 (below): parallel efficiency (C/OpenMP vs. Java/OpenMP)





Graphical user environments for scientific computing

M. Ashworth, R.J. Allan, C.J. Mueller, H.J.J. van Dam, W. Smith, D. Hanlon and B.G. Searle

Many scientific applications now have a Graphical User Interface, or GUI, customised to make its use more intuitive for novices and experts alike. It is interesting to compare technology designed for GUI development and evaluate the kind of functionality commonly incorporated.

GUIs take on a particular importance in the computational Grid world where aspects of managing applications in a heterogeneous distributed environment can be hidden from most users. Indeed projects like UNICORE provide a GUI specifically designed for submitting HPC applications transparently to a range of different systems. Other similar projects provide seamless access to data across a range of file servers or databases.

Our examples are, in the main, taken from work currently ongoing at Daresbury Laboratory in support of the CCP, HPCI and e-Science programmes. We describe implementations in Java, Perl/Tk, Python/Tk and C++/AVS Express and also C and Perl/CGI. There is particular emphasis on using Web browsers as a special form of GUI because of the current focus on the computational Grid and Internet based distributed services (so called Web services).

Technical report available from:
http://www.ukhec.ac.uk/publications/reports/guireport.pdf

Portable application compilation and building for Fortran 90

R.J. Allan and Y.-F. Hu, Daresbury Laboratory

In this report we outline several procedures which have been developed at Daresbury Laboratory to build application software written in Fortran90. In particular we provide examples of how to construct portable makefiles for programs which have modules in several source files or directories. We developed this procedure for porting the CLIPS library.

Some automatic tools, such as makemake, for producing makefiles from source code are described.

We also include illustrations of the choice of compiler flags which have been used on a variety of platforms and comments on the use of cpp for target- and language-specific source lines.

Technical report available from:
http://www.ukhec.ac.uk/publications/reports/build_doc.pdf

Java for HPC

continued

Java to reach a super-linear scaling for the collision stage when the cumulative cache size becomes close to the problem size.

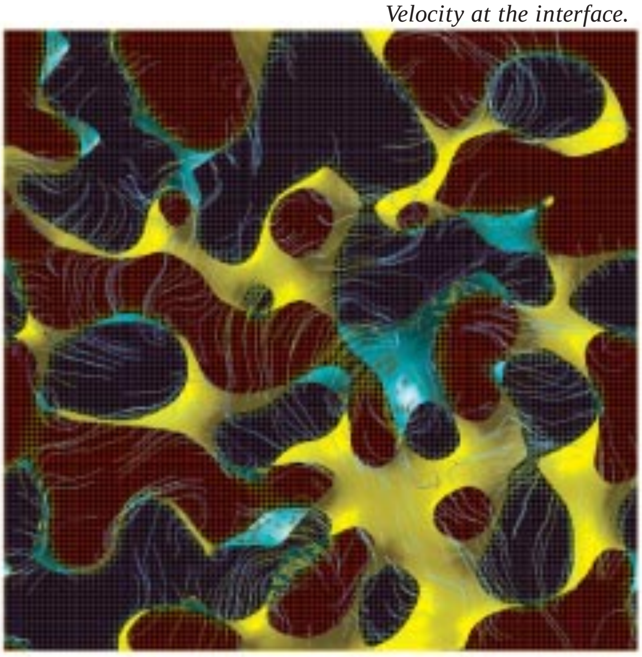
This work is currently being extended to improve the parallel I/O in the Java implementation, to perform further comparisons for larger system sizes ($\geq 256^3$), to carry out further comparisons on ccNUMA architectures and to complete the MPI-Java implementation.

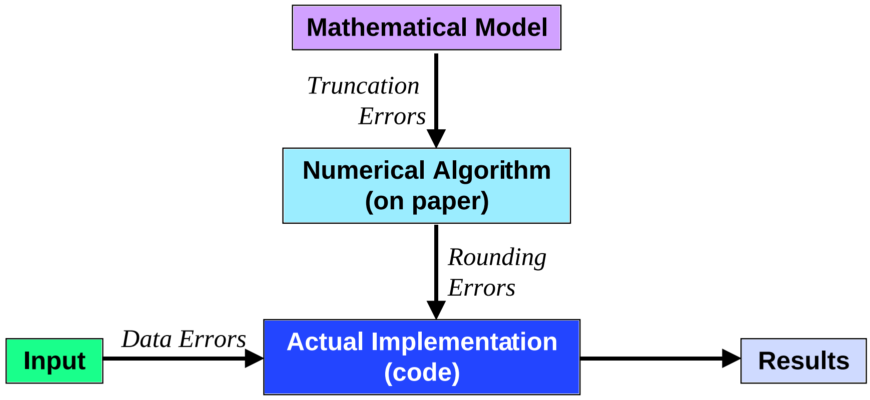
References:

[1] http://www.enacts.org
[2] I. Pagonabarraga, J.-C. Desplat, A.J. Wagner, and M.E. Cates, New Journal of Physics (http://www.njp.org) 3 (9) 2001
[3] http://www.epcc.ed.ac.uk/research/jomp

For further information, see:
http://www.epcc.ed.ac.uk/javagrande

Velocity at the interface.





UKHEC WORKSHOP: Numerical algorithms

Andrew Jackson, EPCC

The UKHEC Workshop on Numerical Algorithms was designed to introduce some of the fundamental issues surrounding numerical computing and algorithm design. It was aimed at those new to scientific computing, but also proved a useful refresher course for more experienced researchers.

The workshop was attended by around twenty post-graduate/post-doctorate participants from a wide range of backgrounds and was well received with those new to the field gaining the most from the experience.

It was structured as follows:

Numeric Computation (Andrew N. Jackson & David Henty)

A mixture of lectures and practical sessions illustrating the limitations of the numerical implementation of real-number arithmetic. This covered not only IEEE floating-point numbers (in some detail) but also interval arithmetic, both of which are now supported by the Sun compiler suite. This allowed the direct comparison of these two paradigms by using both approaches to deal with the same problems.

Algorithm Design & Numerical Libraries (Andrew N. Jackson)

An introduction to the terminology and concepts used in algorithm design. This included a discussion of how numerical analysts can attempt to ensure both accuracy and stability for a wide a range of circumstances as possible.

The problems presented by this process lead naturally to the motivation for the re-use of existing algorithms and, where possible, the re-use of existing code through standard numerical libraries.

Fourier Transforms & FFTW (Gavin Pringle)

As well as giving an introduction to continuous Fourier transforms and to discrete, fast Fourier transforms (FFT's), this lecture also presented the Fastest Fourier Transform In The West (www.fftw.org). This is a self-optimising numerical library that achieves near-optimal performance on an extremely wide range of platforms, thus allowing the user a nearly perfect blend of speed and portability.

Simulating Galaxy Interactions (Prof. Carlos Frenk)

This seminar looked at the ways in which large-scale numerical simulation is being used in the field of cosmology. These studies provide significant insight into our most fundamental theories of galactic interaction and will allow direct comparison with experimental results. The talk also illustrated what may be achieved by combining open implementations of standard algorithms with research ideas and development, through the example of the VIRGO code which itself uses the FFTW library.

IBM SCICOMP 5 workshop

Daresbury, 7-10/5/2002

The 5th workshop of the international IBM Scientific Computing Users' Group, ScicomP5, was held at Daresbury Laboratory on 7-10th May 2002. Some 65 people attended this meeting.

The keynote address was given by Prof. P.J. Durham, the Director of the Computational Science and Engineering Dept., CLRC. This was followed by a number of talks from IBM representatives, looking at issues such as the IBM Scientific and Technical Computing Roadmap, and providing updates on compilers and tools. Throughout the week there was also a number of user presentations, focusing on user experiences of SP systems.

The afternoon excursion on Wednesday was a pathfinder tour of the scenic and historic Roman town of Chester. This was followed by a medieval banquet at Ruthin Castle.

The meeting finished on Friday, with an excellent tutorial session on Power4 Optimisation, led by Bob Blainey and Charles Grassl from IBM.

Sponsors of the workshop were IBM and Pallas.

The lecture material is online on the ScicomP website and may be of interest to readers, see: http://www.spscicomp.org/ScicomP5

UKHEC WORKSHOP: Introduction to XML

Matthew Egbert, EPCC

In January 2002, EPCC ran an Introductory XML Workshop. The workshop was designed to introduce the basic principles of XML and some of the existing XML development tools.

After an investigation into the problems which XML tries to solve and an examination of ‘Meta-Data’, the basic building blocks of XML, (e.g. tags, attributes, elements) were explained. Document Type Definitions (DTDs) were examined, followed by an overview of the two standard methods of parsing XML documents, the Document Object Model (DOM) and the Simple API for XML (SAX). The day concluded with a brief discussion of modern technologies which use XML.

Practicals were interspersed throughout the lectures to fortify the attendees’ understanding of the material. The practicals included writing an XML document from scratch, writing a DTD to describe the legal formats of a given XML document, fixing a SAX parser, and fixing a DOM parser.

The workshop, which took place in EPCC's new training room, was fully booked, and received complementary reviews from many of the attendees.

Further details are available from:
http://www.ukhec.ac.uk/events/xml/

Feedback form

We'd like to know what you think of this newsletter. You can pass on comments by faxing back this form to: 0131 650 6555. This form is also online at: http://www.ukhec.ac.uk/feedback.html

1. Did you find any articles particularly interesting or useful? If so, which?____________________

2. Why?____________________

3. What topics would you like to see covered in future issues? ____________________

4. Please feel free to add any further comments ____________________

Your name ____________________ Job title ____________________

Company/institution ____________________

Address ____________________

Email ____________________

Telephone ____________________ Fax ____________________

Was your copy of this newsletter properly addressed? Yes/No

Contact information

MARTYN GUEST
Computational Science and Engineering Department
Daresbury Laboratory
Daresbury
Warrington WA4 4AD
m.f.guest@dl.ac.uk
01925 603240
http://www.cse.clrc.ac.uk

ALAN SIMPSON
EPCC
JCMB, Maxwell Buildings
University of Edinburgh
Mayfield Road
Edinburgh EH9 3JZ
a.simpson@epcc.ed.ac.uk
0131 650 5120
http://www.epcc.ed.ac.uk

TERRY HEWITT
Manchester Computing
University of Manchester
Oxford Road
Manchester M60 9PL
w.t.hewitt@mcc.ac.uk
0161 275 6095
http://www.man.ac.uk/mrccs
http://www.csar.cfs.ac.uk

MIKE ASHWORTH
Computational Science and Engineering Department
Daresbury Laboratory
Daresbury
Warrington WA4 4AD
m.ashworth@dl.ac.uk
01925 603663
http://www.cse.clrc.ac.uk

DAVID HENTY
EPCC
JCMB, Maxwell Buildings
University of Edinburgh
Mayfield Road
Edinburgh EH9 3JZ
d.henty@epcc.ed.ac.uk
0131 650 5960
http://www.epcc.ed.ac.uk

JON HUNT
Research Facilities Team
EPSRC
Polaris House,
North Star Avenue,
Swindon SN2 1ET
j.hunt@epsrc.ac.uk
01793 444263
http://www.epsrc.ac.uk/hpc

Main UKHEC webpage: http://www.ukhec.ac.uk