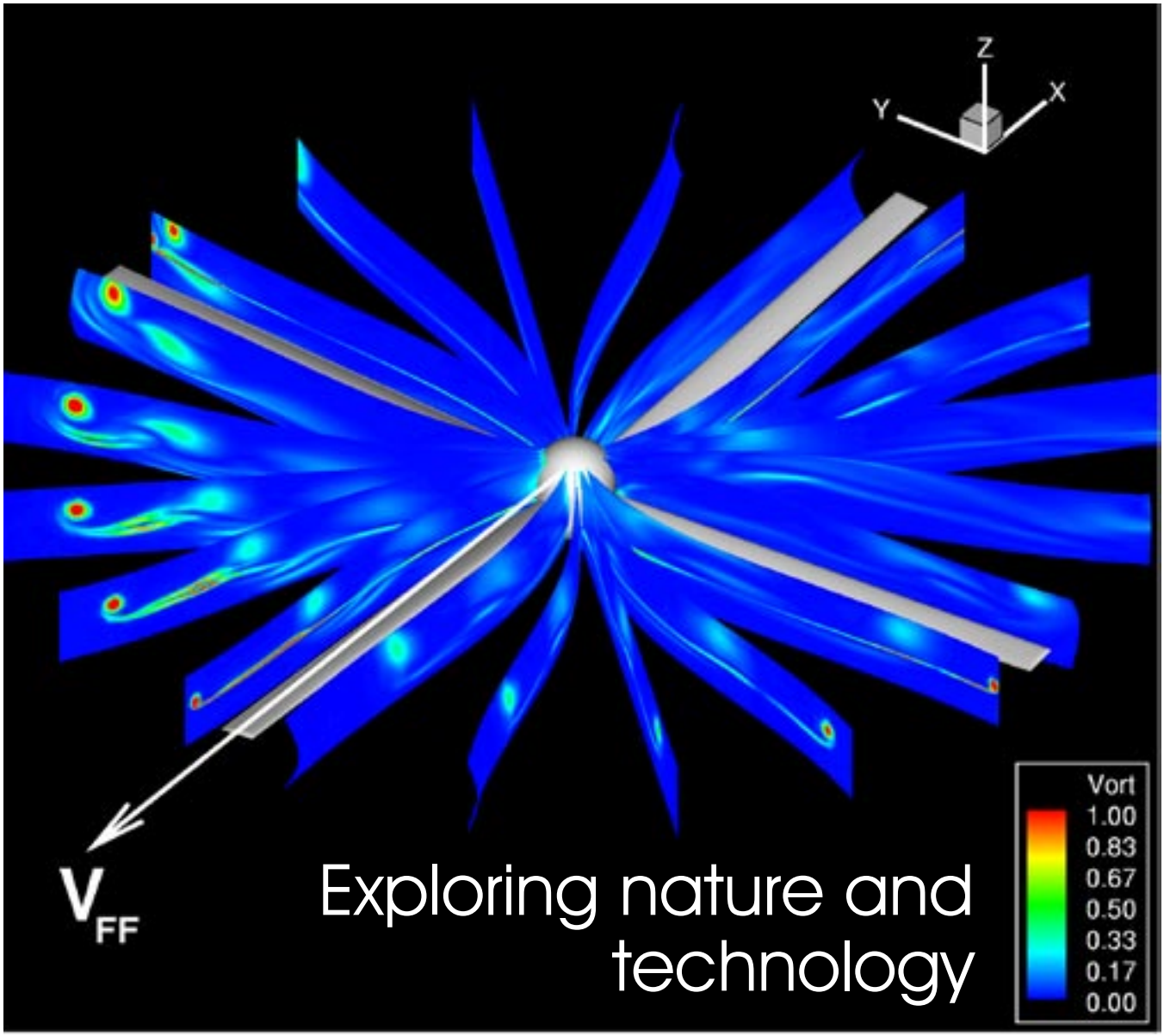
Capability Computing

The newsletter of the HPCx community

[ISSUE 5, SPRING 2005]



CONTENTS

Editorial

Martin Plummer, HPCx Terascaling Team

Hello and welcome to Capability Computing 5, distributed among the HPCx community and also to attendees of ScicomP 11 and SP-XXL, held jointly at Edinburgh in May/June 2005. Our aim in preparing this issue was to provide a flavour of current scientific and technical work on the HPCx service plus a glimpse into the future of UK supercomputing. The three feature science articles describe advances in subjects as diverse as the aerodynamics of helicopter rotor blades, investigations into different solid forms of organic molecules relevant to the pharmaceutical industry and the detailed study of the evolution of laser-driven quantum-mechanical two-electron wavepackets.

We also present two articles which indicate how the performance of scientific codes may be analysed (and hence improved) at the single node 32-cpu LPAR level and the massively parallel capability level. We also show how recent IBM software upgrades have improved capability performance. On the theme of capability performance, we would like to remind you of the Capability Incentives scheme which gives you more computing time for your money.

We present a report on Supercomputing 2004 and a brief look at new technology in the form of field programmable gate arrays. Finally we are pleased to begin the issue with a guest Comment article from Hugh Pilcher-Clayton of the Engineering and Physical Sciences Research Council (EPSRC). HECToR, the High End Computing Terascale Resource, is the proposed 'next generation' (beyond HPCx and the CSAR service at Manchester) national high performance computing service.

The future of UK supercomputing

Hugh Pilcher-Clayton, EPSRC Programme Manager for High-End Computing

The nature of research in science and engineering has changed over the past two decades – from an activity based almost entirely on theory and experiment, to one based on theory, experiment and computation in comparable measure. Computational simulations are now indispensable for the numerically oriented fields of research, such as climatology, earth sciences, particle physics, cosmology, astrophysics, chemistry, materials, fluid dynamics, atomic and molecular physics, plasma physics, nanoscience and biomolecular sciences. The increasing sophistication of mathematical models and complexity of simulations, along with continuing improvement in computing power, has made it possible to address new scientific questions, which in turn increases the demand for greater computational resources.

The fundamental principle of the UK's long-standing strategy, as re-stated in the Strategic Framework for High End Computing issued by the High End Computing Strategy Committee (HSC) in June 2003, is to provide national general-purpose high performance computing systems at a world-class level, with more specialised systems to address particular scientific questions. This strategy has evolved with time, reflecting the changing requirements of the scientific community, the development of the technology and the need to ensure value for money. Greater emphasis is now placed on capability rather than capacity computing and on the establishment of a service which includes not only the system itself but the facilities management of that system and the provision of Computational Science and Engineering (CSE) support.

There has been much debate on the definition of capacity and capability computing! My own attempt to articulate the difference is as follows. Capacity computing is sufficient for the high throughput of a large number of simulations, each of which addresses a relatively small problem size. It can be provided using the distributed computer infrastructure at departmental and university level, access to which will be enabled by the emerging e-Infrastructure. Capability computing, on the other hand, is needed for simulations that address much larger problem sizes. Such simulations, which require high communication bandwidth and low latency times, can only be achieved in a viable timescale by a high performance system and not by using a distributed computing architecture.

Of crucial importance is not the theoretical peak performance of a system or its sustained performance according to Linpack, but the sustained performance of scientific codes. CSE support is therefore vital, and accordingly its level was increased for the HPCx service, with a further increase planned for the proposed HECToR service. For HECToR there will also be much greater use of a suite of scientific codes to evaluate the performance of the system, supplemented with industry standard benchmarks from the HPC Challenge benchmark suite. A personal aspiration is not to run the Linpack benchmark at all, as I consider it to be a very poor indicator of the performance of scientific codes and hence highly misleading, but I suspect that it will be politic to do so in order to provide an assessment of HECToR's ranking within the world.

The procurement process for HECToR is proceeding well, but because of financial constraints the planned start date of the 6-year service has been delayed until April 2007. A decision on a partnership with the Met Office will be made in July 2005 and the OJEU notice issued at the beginning of November 2005.

Because of its integrated strategy, the CSAR and HPCx services, and plans for the HECToR service, the UK is recognised within Europe as being a key player in High End Computing. EPSRC, representing the UK, has been involved in discussions with initially the French and Germans, but with the intention of including other countries as well, about the establishment of a European petascale high performance computing infrastructure during the timescale of the 7th Framework Programme of the European Commission. Clearly, these are early days for this initiative, but if it is successful then Europe should have a high performance computing capability that enables it to compete scientifically world-wide.

Field programmable gate arrays

Ian Bush, HPCx Terascaling Team

One of the major new themes in high performance computing (HPC) is the rapidly growing interest in the use of field programmable gate arrays (FPGAs). This has been driven by the recognition that current HPC architectures, such as HPCx or Newton, face considerable problems in scaling to the performance required to address the future requirements of computational science, especially in terms of price and power consumption. It has been observed that in the not too distant future the power requirements of a top-end HPC machine will exceed what the National Grid provides to a major city like Manchester!

FPGAs are one possible solution to these problems. An FPGA is effectively a programmable chip. Each consists of typically many millions of gates that can be connected in an almost arbitrary fashion, thus allowing any program to be implemented in hardware. Though they are clocked slowly compared to typical modern CPUs, a few hundred MHz instead of a few GHz, they are, in principle, capable of performing many more operations per clock cycle. This high degree of parallelism coupled with the falling price of FPGAs means that their price performance is becoming attractive, and further the power consumption is low when compared with a modern CPU.

At present the use of FPGAs in HPC is limited. The main reason for this is that implementing 64 bit IEEE floating point arithmetic requires a lot of gates, so limiting the degree of parallelism on the FPGA (as you can only have a few pipelines on the FPGA) and so making the performance unattractive due to the slow clock rate. However, the next generation of FPGAs will both have many more gates and also clock at a somewhat higher rate; hence the growing interest in the HPC community exemplified by both Cray and SGI announcing systems that contain FPGAs.

As such CSAR held a symposium on 21-22 February to discuss current and possible future uses of FPGAs in HPC. The meeting covered a very wide range of areas, from the technical specifications of FPGAs, through systems and programming tools to current applications of FPGAs. Cray presented their XD1 system, where the FPGA can act as a co-processor in an Opteron based system, and SGI their SA brick which can be incorporated into an Altix system, sitting directly on the NUMAlink interconnect. A variety of programming paradigms were presented, ranging from GUI interfaces for dragging and dropping components into the desired network, through C-like languages to low level methods. While many of the more impressive applications of FPGAs were based on integer or fixed point arithmetic, such as cryptography, some were using floating point though typically not full 64 bit IEEE.

More details can be found at:
http://www.csar.cfs.ac.uk/services/courses/fpga_timetable.shtml

---

HPCx Capability Incentives update: get more CPU time for your money!

M Plummer, HPCx Terascaling Team

The Capability Incentives scheme was introduced to encourage efficient massively parallel calculations that exploit HPCx to the full (see *Capability Computing* issue 2, http://www.hpcx.ac.uk/about/newsletter). Discounts are awarded for codes that scale with a wall-clock time speed-up of 1.7 or more from 128-256 processors (bronze award), 256-512 processors (silver award) and 512-1024 processors (gold award). Award holders receive a discount code to be included in their LoadLeveler scripts, resulting in fewer AUs than are actually used being deducted from the project budget. Current discounts are 5% for a bronze award, 15% for a silver award and 30% for a gold award.

To apply for a discount, send details of your code, your test case and timings to the helpdesk. Our only caveat is that the test case should be a representative scientific run of your code.

CODE	RATING
LB3D	Gold
Ludwig	Gold
NAMD	Gold
PDNS3D	Gold
CRYSTAL	Silver
LAMMPS	Bronze
GAMESS-UK	Gold
ROTORMBMGP	Gold

The figure shows a selection of current award-holding codes. By the time you read this, more codes may have been added to the list. More information on the Capability Incentives scheme may be found at http://www.hpcx.ac.uk/services/policies/capability.html. The HPCx Terascaling Team works on general and parallel optimization of a selection of scientific codes each year, based on grant provision and general importance of the code within the HPC community. This work is set out in the HPCx Annual Plan as agreed with EPSRC.

Kinetic insights into polymorphism in pharmaceutical (solid state) chemistry

S Hamad and C R A Catlow, Royal Institution, London UK; A T Hulme and S L Price, University College London, UK.
Professor Sarah (Sally) Price is principal investigator of the Basic Technology Project 'Control and Prediction of the Organic Solid State.'

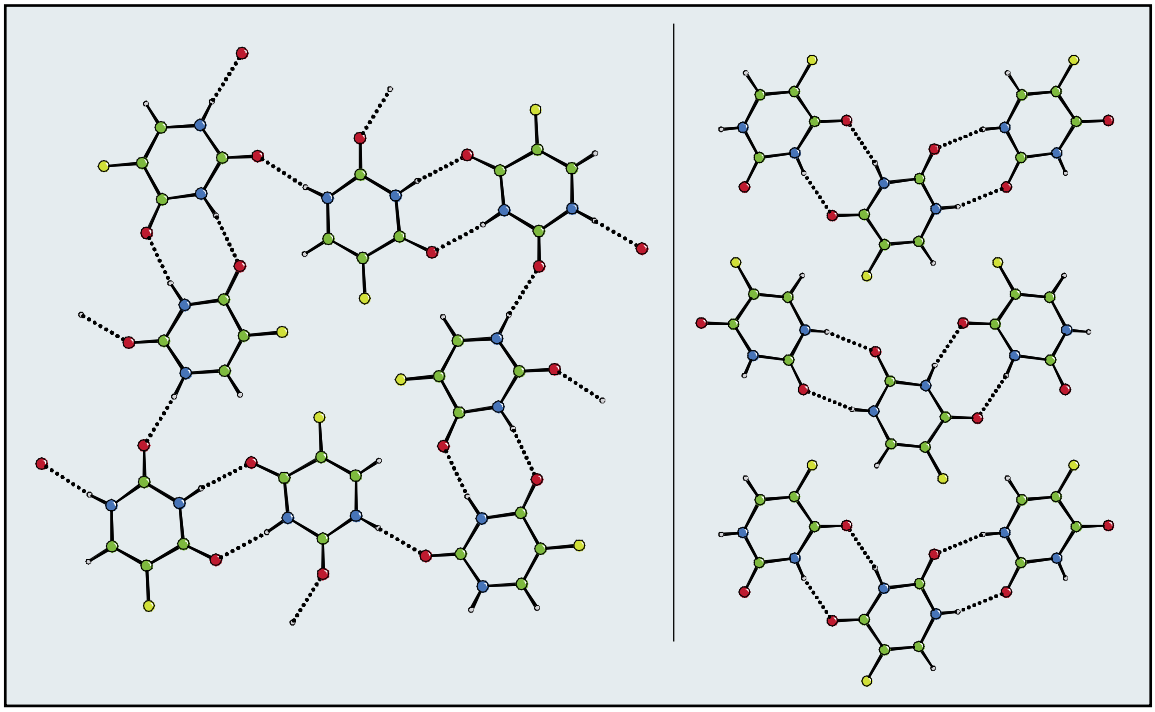


Figure 1. The sheets in the crystal structures of form I (left) and form II (right) of 5-fluorouracil ($C_4H_3FN_2O_2$).

Why can a new solid form of an organic molecule only be found after many man-years of work on the compound?

The late discovery of new polymorphs, which are often more thermodynamically stable, can lead to problems in producing the original solid form in a controlled manner. This is a major problem for the pharmaceutical companies, who are only licensed to sell their products in a specified solid form, as although polymorphs are chemically identical, their physical properties, such as dissolution rates, differ. As recently as 1998, Abbott Laboratories had to reformulate their anti-HIV drug ritonavir, when the manufacturing process suddenly started to produce a more stable polymorph [1]. The discovery of a new polymorph by a rival pharmaceutical company, even if it is metastable but can be controllably produced, is a threat to the company's patent on their active molecule. Thus an ability to predict polymorphism computationally would be of considerable practical utility to the pharmaceutical and other organic materials industries, as well as aiding the design of new materials.

We have recently found [2] a new polymorph of 5-fluorouracil, an anti-cancer agent that has been known since 1957. The original crystal structure (now form I) is unusual, with 4 molecules in the asymmetric unit cell and close contacts between the fluorine atoms (Figure 1). A computational search for minima in the lattice energy, calculated using a realistic model for the electrostatic intermolecular interactions, found that there were many hypothetical structures with lattice energies up to 6 kJ/mol more stable than the known form, and more typical hydrogen bonding motifs. A large series of crystallisation experiments finally yielded a crystal of a new polymorph, form II, which was shown by X-ray diffraction to have the structure predicted as the global minimum in the lattice energy within a few percent error in the cell lengths.

It proved remarkably difficult to obtain further samples of form II, until it was realised that the solvent from which it was crystallised, nitromethane, had to be dry. The hydroscopicity of nitromethane and the low solubility of 5-fluorouracil imply that there would be between 4 and 40 water molecules to each 5-fluorouracil in a solution that had been exposed to normal air. The specificity of the solvent required for the production of form II suggested that the kinetics of molecular association within different solvents might explain the polymorphism of 5-fluorouracil.

HPCx was used to investigate [3] this hypothesis, by performing a series of Molecular Dynamics simulations, using DL_POLY [4], of 5-fluorouracil in water and nitromethane. It was observed that water molecules hydrogen-bonded strongly to the carbonyl

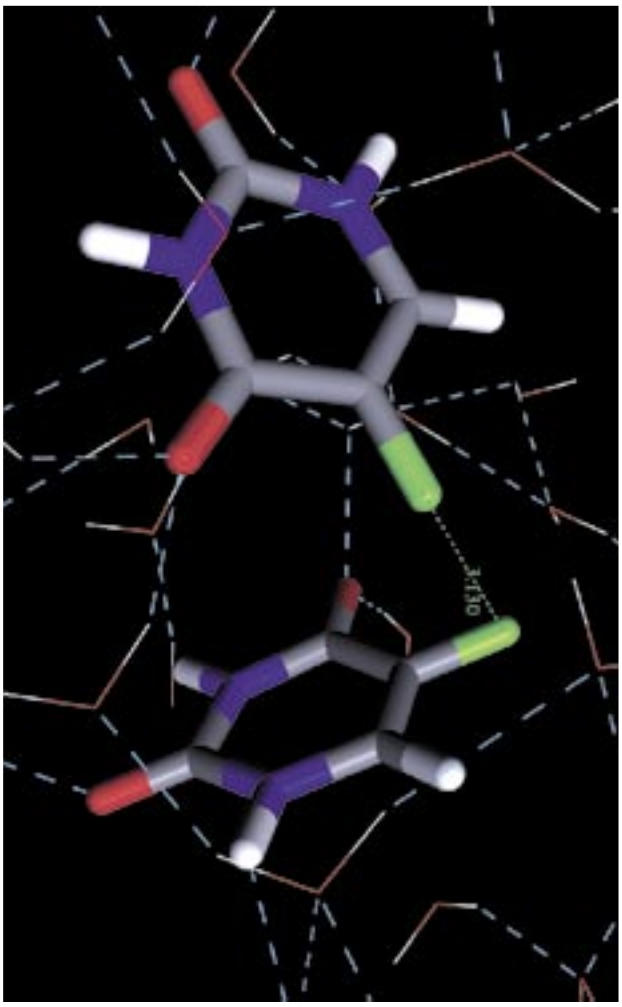
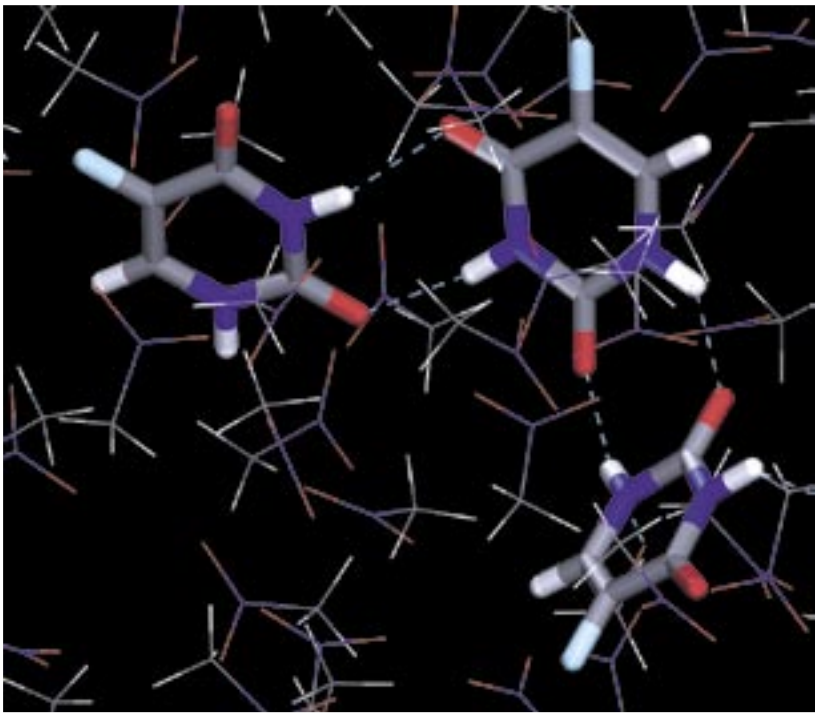


Figure 2 (left). Snapshot from the MD simulation of 5-flurouracil in water, showing the hydrogen bonds within the solvent and to 5-fluorouracil, and the short F···F contact in the solute association.

Figure 3 (below). Snapshot from the MD simulation of 5-fluorouracil in nitromethane, showing the hydrogen bonding that is also seen in form II.



groups, forming a partial hydration sphere around the polar part of the molecule, but not approaching the fluorine closely. Thus there was a tendency for the 5-fluorouracil molecules in water to associate through close F···F contacts as shown in Figure 2. Some single hydrogen bonds were also formed in aqueous solution, but the hydrating waters seemed to prevent the formation of a second hydrogen bond between the solute molecules. In contrast, nitromethane molecules were more evenly and loosely associated with the 5-fluorouracil molecules in the simulated nitromethane solution. In this case, the contacts between a pair of 5-fluorouracil molecules generally formed two hydrogen bonds quickly and this dimer motif was very persistent. Indeed, some instances of trimer formation were also observed, such as that shown in Figure 3, which forms the building block for form II. Thus the MD simulations seem to account for the different crystal forms in terms of the differences in the molecular association in the different solvents, in a way that has also been recently inferred from FTIR solution spectroscopy to account for the polymorphism in tetrolic acid [5].

These simulations were only designed to study the very first step in crystallisation, the initial association of the solute molecules. Following this success, we hope to use the HPCx capability to give molecular level insights further into the nucleation process for other organic systems, but this will require even larger system sizes and simulation times.

This Molecular Dynamics investigation was carried out by Said Hamad under the direction of Richard Catlow at the Royal Institution, with the technical support of Maurice Leslie, CCLRC Daresbury.

The work was done in close collaboration with the experimental work of Ashley Hulme and Derek Tocher at University College London Chemistry Department, as part of the Basic Technology project 'Control and Prediction of the Organic Solid State', funded by the Research Councils UK, whose principal investigator Sarah (Sally) Price wrote this article.

If you are interested in the project's progress, please see the website http://www.cposs.org.uk *which includes an invitation to an open day on 13th September.*

Reference List

[1]. Chemburkar, S. R.; Bauer, J.; Deming, K.; Spiwek, H.; Patel, K.; Morris, J.; Henry, R.; Spanton, S.; Dziki, W.; Porter, W.; Quick, J.; Bauer, P.; Donaubauer, J.; Narayanan, B. A.; Soldani, M.; Riley, D.; McFarland, K. *Organic Process Res. Dev.* 2000, 4, 413-417.

[2]. Hulme, A. T.; Price, S. L.; Tocher, D. A. *J. Am. Chem. Soc.* 2005, 127, 1116-1117.

[3]. Hamad, S.; Moon, C.; Catlow, C. R. A.; Hulme, A. T.; Price, S. L. *in preparation*.

[4]. Smith, W.; Forester, T. R. *J. Mol. Graphics* 1996, 14, 136-141.

[5]. Parveen, S.; Davey, R. J.; Dent, G.; Pritchard, *R. G. Chem. Commun.* 2005, 1531-1533.

Single node performance on HPCx

J Mark Bull, HPCx Terascaling Team

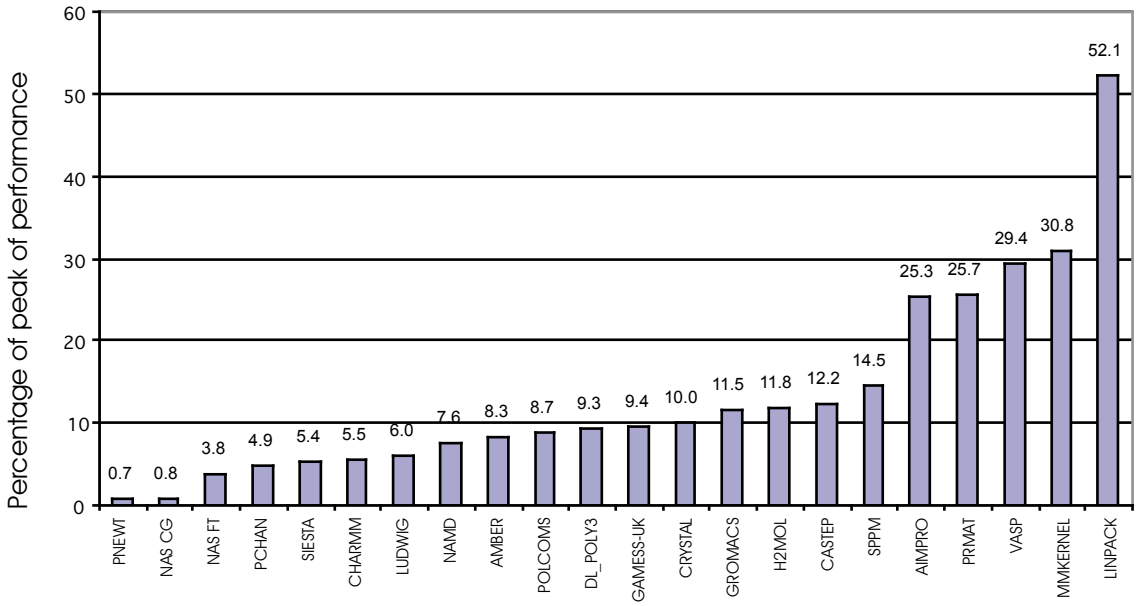


Figure 1. Percentage of peak performance achieved.

The overall performance of an application code on HPCx can be thought of as a combination of two factors: how well the code performs on a single node (32 processors) and how well the performance scales with increasing numbers of nodes. This article concentrates on the former aspect: we have measured the performance of a range of benchmarks and user application codes on a single node of the HPCx Phase 2 system. We consider primarily the floating point performance of these codes and how this relates to the peak floating point performance of the hardware. In addition to measuring floating point performance, we also use the hardware counter facilities of the Power4+ processors to record other metrics for each code.

In this study we used parallel versions of all the codes and collected data (with one exception) from runs using all 32 processors of a p690+ shared memory node. We considered this preferable to running sequential codes because in serial mode a single CPU of a p690+ node has access to one entire Level 2 cache and all the Level 3 cache of the node. The vast majority of production work on the system is carried out with parallel codes where all 32 processors of each node are utilised. Sequential codes running in isolation can therefore achieve inflated performance that is unrepresentative of the way the system is used in practice.

The 22 codes we tested were:

- Benchmarks: LINPACK, MMKERNEL, NAS CG, NAS FT, SPPM.
- CFD Applications: PCHAN, PNEWT
- Environmental Modelling Applications: POLCOMS
- Molecular Dynamics Applications: AMBER, CHARMM, DL_POLY3, GROMACS, NAMD.
- Physics Applications: H2MOL, LUDWIG, PRMAT
- Quantum Chemistry and Materials Science Applications: AIMPRO, CASTEP, CRYSTAL, GAMESS-UK, SIESTA, VASP.

Further details of the code can be found in [2].

hpmcount

`hpmcount` is a utility which gives access to the hardware counters on the IBM Power4+ processor [1]. In addition to the counter groups 5, 56 and 60 described in [1], we also used counter group 40, which gives information about prefetch streams, ie when data is prefetched to Level 1 and Level 2 cache (for details, see [2]).

Methodology

Each benchmark or application was run on 32 processors (for reasons which relate allowed processor numbers to symmetry in the problem geometry, H2MOL was run on 28 processors). For each code, four separate runs were made, instrumented with `hpmcount` and recording events from counter groups 5, 40, 56 and 60.

For each code, a dataset and/or input parameters were chosen to yield a problem size which was small enough to fit in the 27Gb of available memory on the node, but large enough to be

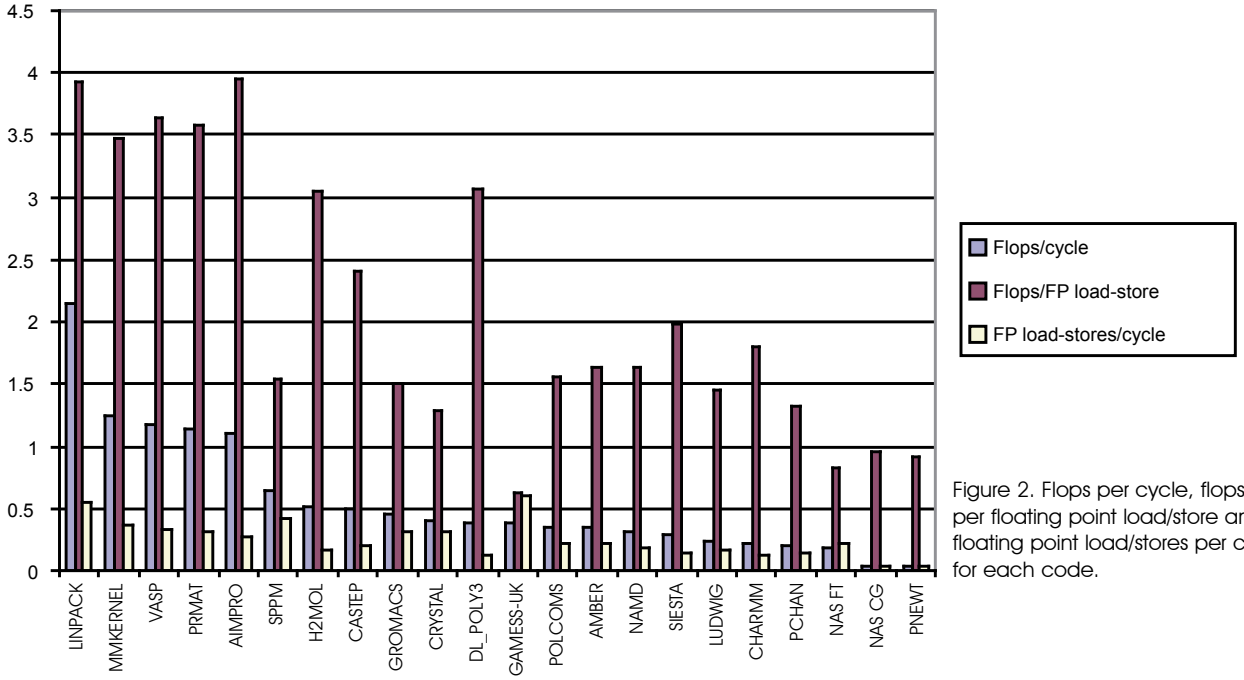


Figure 2. Flops per cycle, flops per floating point load/store and floating point load/stores per cycle for each code.

representative of real usage of the application codes, and also large enough so that communication overheads did not dominate the execution time. Actual scientific application runs will of course give variations in performance (both better and worse) depending on the overall job size and nature, total number of processors, effective available memory and the efficiency of the parallelisation scheme across nodes. However, the test results provide very useful data for non-subject-specialist optimisers to interact effectively with code developers in order to improve general performance of the codes.

Floating point performance

Using hardware counter group 60, `hpmcount` reports the flop rate for each process, based on wall clock execution time. For each code, the total flop rate for all 32 processors was calculated, and expressed as a percentage of the peak flop rate for the p690+, which is 217.6 Gflop/s. This data is shown in Figure 1.

The highest performance (just over 52% of peak) is, not surprisingly, attained by the Linpack benchmark. A small group of codes (MMKERNEL, VASP, PRMAT and AIMPRO) achieve between 25% and 31% of peak performance. All the codes in this group make extensive use of tuned dense linear algebra libraries for a significant portion of the computation. Most other codes achieve between 3% and 15% of peak performance, while two codes (NAS CG and PNEWT) achieve less than 1% of peak. These latter two codes are characterised by highly irregular data access patterns.

Other metrics

In addition to the flop rate, `hpmcount` provides a large number of other metrics calculated from raw hardware event counts. These are described in detail in [1] and section 4 of [2], together with additional useful user-derived metrics similarly obtained from the `hpmcount` raw data. In [2] we assess the importance of these individual metrics in determining the overall flop rate by calculating correlation coefficients across all 23 codes.

The metrics with the highest correlation to flop rate are *flops per floating point load/store*, the ratio of floating point operations to the total number of floating point load and store instructions (called Computational Intensity in `hpmcount` output) and *floating point load/store rate*, the rate of issue of floating point load and store instructions. This is not surprising, since the flop rate can be written as the product of *flops per floating point load/store* with *floating point load/store rate*. Figure 2 shows how these two metrics combine to give the resulting flop rates. Note that in Figure 2, for ease of presentation, the flop rate is given in flops per cycle, and the load/store rate in operations per cycle.

The five codes with the highest flop rate all execute between 3.5 and 4 flops for every floating point load/store. Some codes with modest performance also have high values of flops per floating point load/store, but low floating point load/store rates (eg DL_POLY). For others, the converse is true (eg GAMESS-UK). The codes with worst performance execute less than one flop for every floating point load/store, and also have low floating point load/store rates.

Analysis of other metrics, additional details of the codes, and further data can be found in [2].

Acknowledgements

The author gratefully acknowledges the assistance of the following people in collecting data for this study: Jonathan Followes of IBM; Martin Plummer, Ian Bush, Paul Sherwood, Andy Sunderland, Huub van Dam and Mike Ashworth of CCLRC Daresbury Laboratory; Joachim Hein, Gavin Pringle, Kevin Stratford, Fiona Reid and Lorna Smith of EPCC, The University of Edinburgh.

References

[1]. Hein, J, *Using the Hardware Performance Monitor Toolkit on HPCx*, HPCx Technical Report HPCxTR0307, available from: http://www.hpcx.ac.uk/research/hpc/

[2]. Bull, M, *Single Node Performance of Applications and Benchmarks on HPCx,* HPCx Technical Report HPCxTR0416, available from: http://www.hpcx.ac.uk/research/hpc/

Inter-node communication on HPCx: the impact of IBM software upgrades on the High Performance Switch

Martin Plummer, with data prepared by the HPCx Terascaling Team.

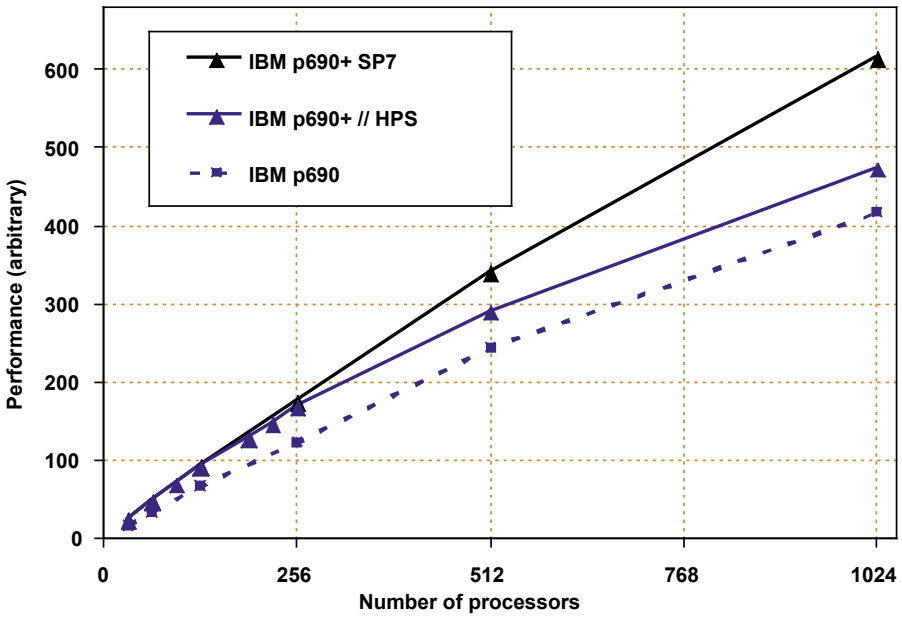


Figure 1.
Performance of NAMD: F_1-ATPase benchmark

The HPCx Phase2 system came into full user service on 29 April 2004 as reported by Steve Andrews in *Capability Computing* issue 3 (http://www.hpcx.ac.uk/about/newsletter). The Phase2 upgrade (effectively a completely new system) doubled the overall capability of HPCx with a larger number of more powerful processors and the introduction of IBM's new High Performance Switch (HPS, formerly known as 'Federation') together with appropriate new operating system software. HPS has greatly increased bandwidth and reduced message latency compared to the old 'Colony' switch of HPCx Phase1.

Following the introduction of the HPS switch there have been important upgrades to the HPS software. The first took place on 28 July 2004 as part of an upgrade to AIX known as Service Pack 7 (SP7) and delivered significant improvements to the point-to-point message latency and the message transfer times, especially in the mid-range of message lengths (around 64kB). The second upgrade with Service Pack 9 (SP9) had less effect on MPI performance but contained a significant performance improvement to some LAPI functions, affecting those codes which call LAPI directly, for example via the Global Array tools. SP9 and succeeding upgrades also provide important safety features in the event of systems failures, which general users should ideally never actually knowingly encounter, as well as other background features. By the time you read this article SP12 will be installed, as well as various runtime library updates which should, for example, provide some enhancement of OpenMP performance.

In this article we would like to indicate improvements in scientific application performance mainly due to the HPS and SP7. Full details are available in HPCx Technical Report TR0417[1] including references for all the technical terms above and from which the figures are taken. We also refer to Mark Bull's article in this issue on single node performance (see page 6).

The HPS and its software provide the all-important communications between the 32-CPU single nodes or LPARs and are crucial for efficient capability computing and terascaling. The HPCx Terascaling Team regularly benchmarks a large range of application codes ranging from quantum physics to chemistry and biochemistry, materials, engineering and environmental science. Various codes are selected for optimisation each year according to

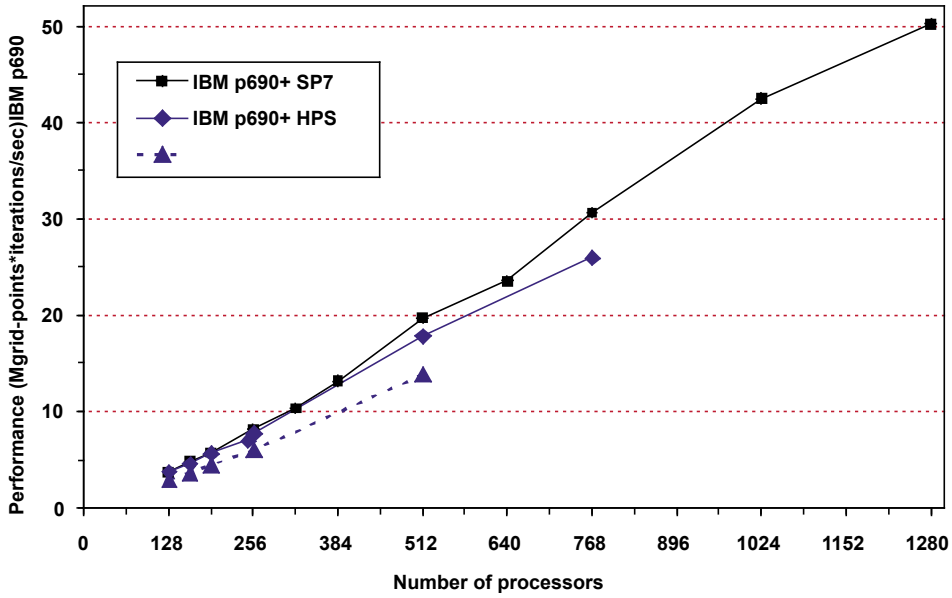


Figure 2. Performance of PCHAN: T3 (360x360x360) benchmark.

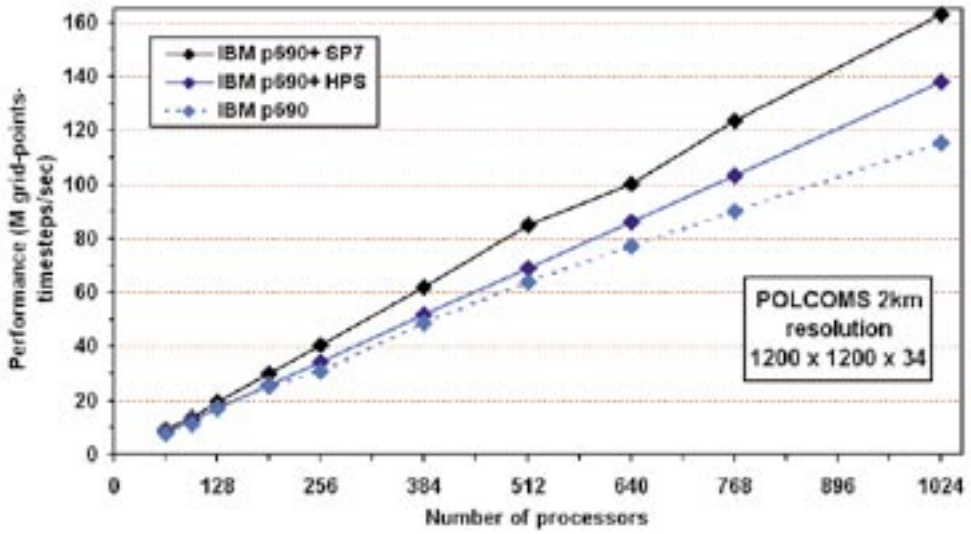


Figure 3. Performance of POLCOMS: 2km coastal simulation (1200x1200x34 grid points) benchmark.

the HPCx Annual Plan as agreed with the EPSRC.

Figure 1 shows performance of the molecular dynamics code NAMD designed for simulations of large bio-molecular systems. Figure 2 shows performance of the computational fluid dynamics code PCHAN for a turbulent flow benchmark. Figure 3 shows performance of the environmental code POLCOMS running a coastal hydrodynamic simulation. Full details of the codes and test cases are given in [1]. In each case the label p690 refers to HPCx Phase1, the label p690+ HPS refers to HPCx Phase2 before the Service Pack upgrades and the label p690+ SP7 refers to HPCx Phase2 with SP7.

All three codes show good scaling which becomes excellent scaling following the Service Pack upgrade. This performance improvement for runs with large numbers of nodes (LPARs) is seen across the range of codes tested in [1] including workhorse parallel library routines such as ScaLapack diagonalisers. Another Technical Report [2] provides a detailed analysis of improvements to basic communication operations in MPI and LAPI from the SP7 and SP9 upgrades.

We note that, arguably, our three example codes all seemed to be underperforming in the single node tests in Mark's article. In fact, they are very carefully designed to run at capability levels with minimal communication costs. It is the job of the Terascaling Team to work with code developers to maximise performance at the single node level while maintaining and/or enhancing inter-node capability communication efficiency ([3], for example, describes very recent optimisation work on POLCOMS).

References

[1]. M Ashworth, I J Bush, M F Guest, M Plummer, A G Sunderland and J Hein, *HPCx Technical Report HPCxTR0417*: http://www.hpcx.ac.uk/research/publications

[2]. A Jackson, *HPCx Technical Report HPCxTR0412*: http://www.hpcx.ac.uk/research/publications

[3]. M Ashworth, J Holt and R Proctor, *HPCx Technical Report HPCxTR0415*; M Holden, *HPCx Technical Report HPCxTR0502*: http://www.hpcx.ac.uk/research/publications

Helicopter rotor wake simulation using HPCx

C B Allen, Department of Aerospace Engineering, University of Bristol, UK

Dr Christian Allen is Reader in Computational Aerodynamics and Head of the Aerodynamics group in the Department of Aerospace Engineering, University of Bristol and a current EPSRC Advanced Research Fellow in the field of rotary wing simulation and optimization.

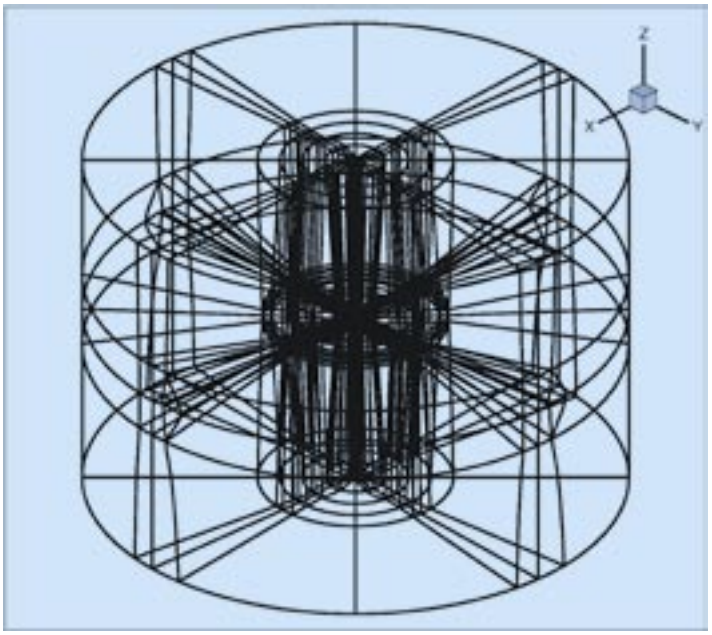


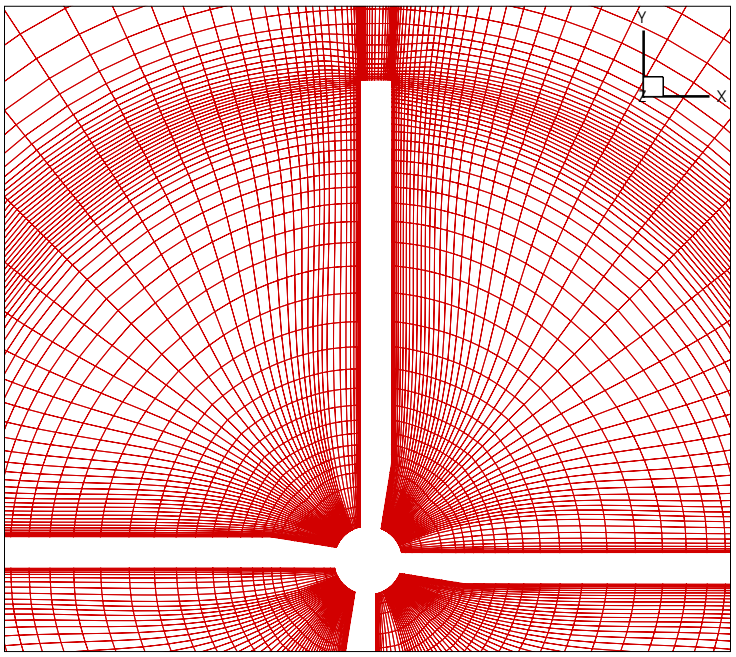
Figure 1. Computational domain and block boundaries for a 408 block grid, for the four-bladed case.

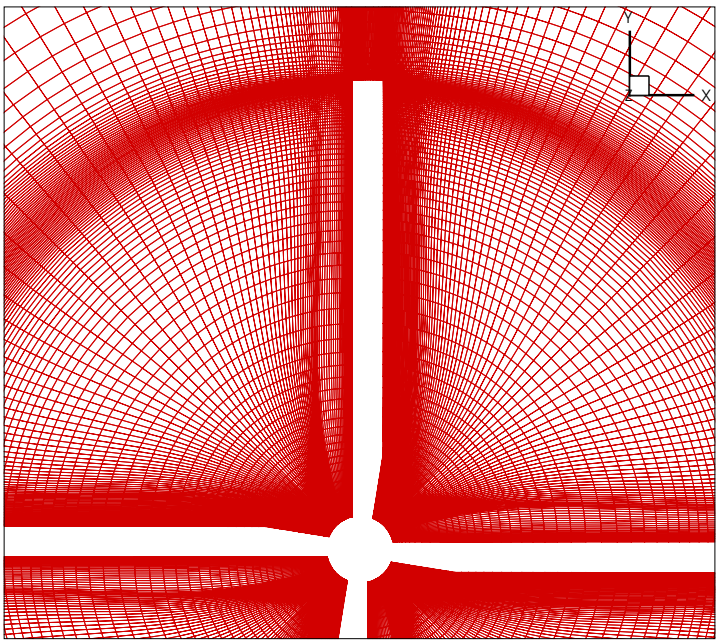
The aerodynamics of helicopter rotor blades is one of the most interesting and challenging problems facing aerodynamicists. Predicting the flow around a blade is a totally different problem to an aircraft wing, which has an (almost) uniform constant speed flow over it, and what happens downstream of the wing is not that significant. That is not the case for a rotor blade, where the speed of the blade varies from very low at the root, to very high at the tip, and in forward flight the effective velocity the blade sees is different at every point around the azimuth (revolution). Hence, the flow is highly three-dimensional and unsteady. Furthermore, and most significantly, each blade moves into fluid that has already been disturbed by the previous blade(s). The loud buzzing, vibrating, sound caused by helicopters is due to the wake, and particularly the strong vortex from each blade tip, shed from one blade being hit by the following one. The accurate prediction of this blade-vortex interaction (BVI) is essential for both civil helicopters, to attempt to reduce 'noise pollution', and military, to avoid detection.

Until recently, numerical simulation of these flows was rare, due to excessive cost, but rapidly increasing computer power and code capability have meant this is now possible.

Computational methods for fluid flow simulation involve filling the physical domain of interest with a computational grid, ie filling the domain with a number of cells, each one having the solution stored in it. The solution is the local values of flow variables, ie density, pressure, velocity, etc. Using applied techniques from mathematics and physics, methods can be developed which march the solution forward in time, starting from an initial guess and iterating until the solution converges (stops changing), and these methods can be used to simulate steady or unsteady flows. The numerical approximations have a truncation error associated with them, ie the difference between the real equations and the numerical approximation, and the error is a function of the grid spacing. Hence, the finer the computational grid used, the more accurate the solution is. The error in the numerical approximation is known as dissipation, and so the coarser the grid the more dissipation is added to the flow and, hence, the interesting parts of the flow are 'smeared out'.

This leads to the major problem with simulation of rotor flows. The aerodynamicist's job is to predict forces on wings or blades, and to do this the flow on the surface is required. For a fixed-wing case, grid points can be clustered close to the surface so the flow is computed accurately there, and the grid away from the surface can be coarse as the flow there is of little significance. However, as





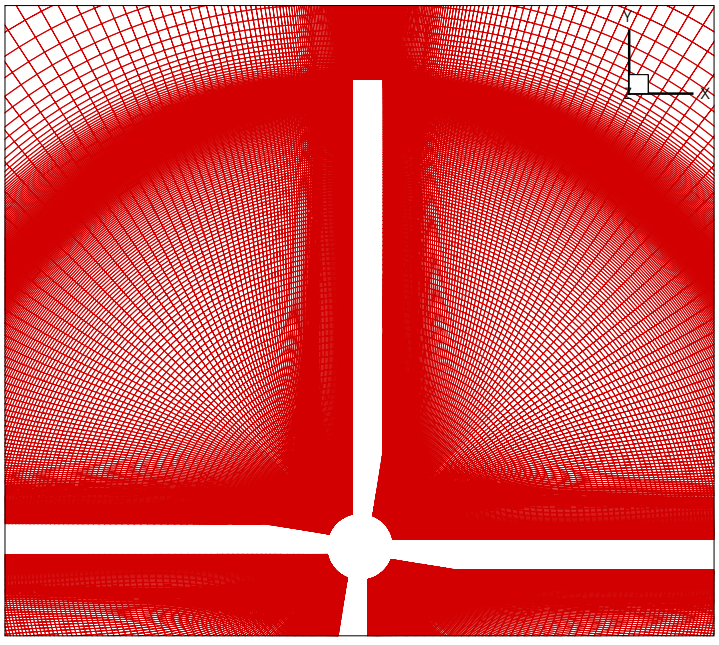
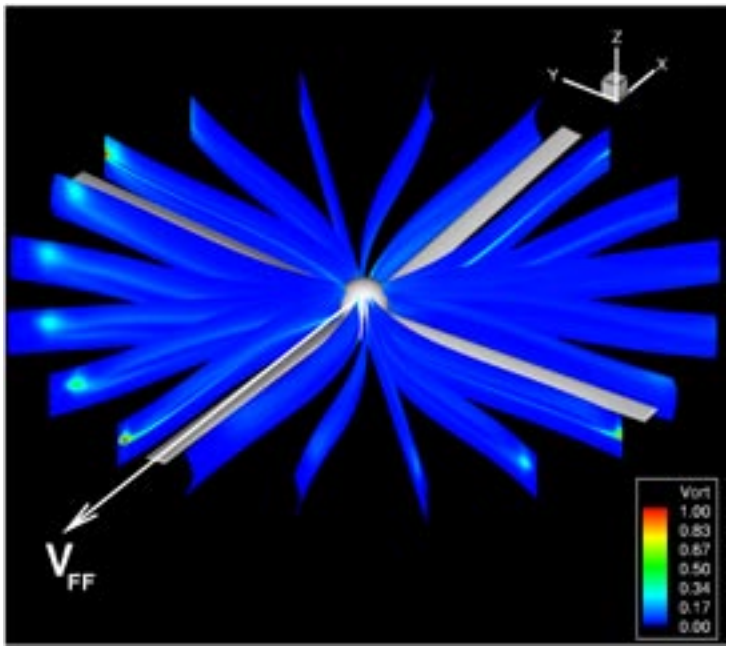


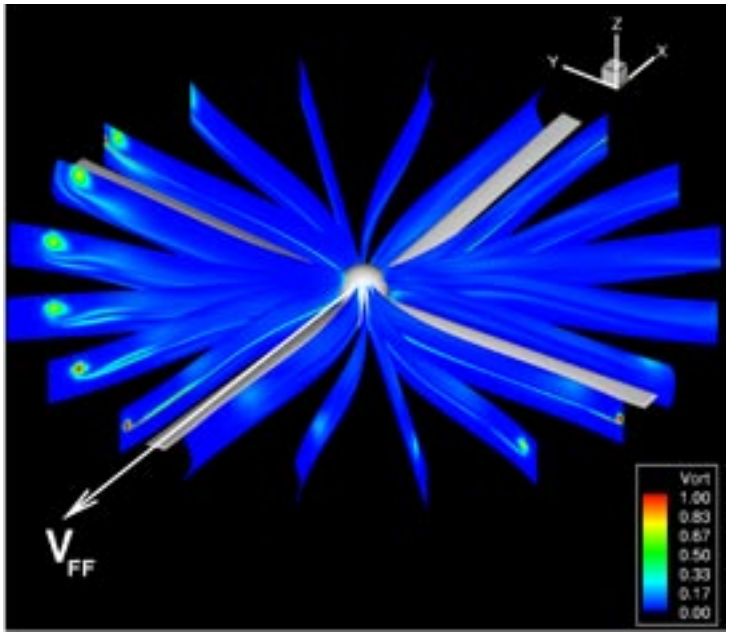
Figure 2. Near-hub grid distribution in rotor disk for the four-bladed case, for 1, 8 and 32 million points respectively.

mentioned above, the flow around a rotor is significantly affected by the wake from previous blades. Hence, to simulate rotor flows requires extremely fine meshes throughout the grid, otherwise the wake from each blade is dissipated by the numerical scheme before it hits the next one. Furthermore, to capture the wake over many turns, particularly for hovering rotors where a helical wake develops beneath the blades, requires a large number of iterations/time-steps. Hence, rotor flow simulation requires many time-steps on a very fine mesh, and this leads to huge run times.

Flow-solver and aspects of parallelisation

Dr Allen has developed both flow-solver and grid generation methods for rotary wing applications. The code is a structured multiblock, upwind solver, using implicit time-stepping for unsteady calculations, with explicit-type time-stepping within each real time step. Multigrid acceleration is used to improve convergence. The multiblock approach adopted is ideal for parallelisation, and the code has been parallelised using MPI. The code has been written so that for each grid block it only requires the IMAX, JMAX, KMAX dimensions, the IMAX×JMAX×KMAX coordinates, and one line per block face listing the boundary condition flags. Hence, each block can be written to a separate file, and a header file lists the name of each block file. All connectivity





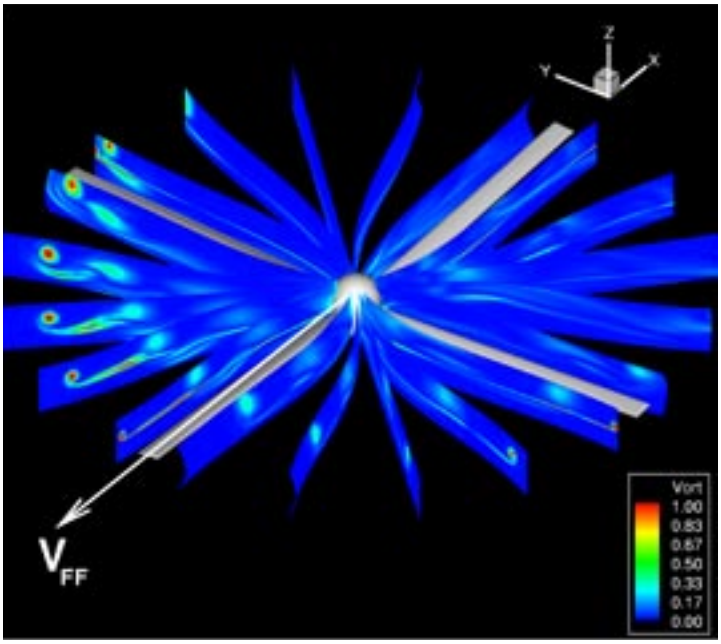


Figure 3. Vorticity shading in selected grid planes for 1, 8 and 32 million points (left, middle and below) respectively.
Vorticity is a measure of local rotation in the fluid.

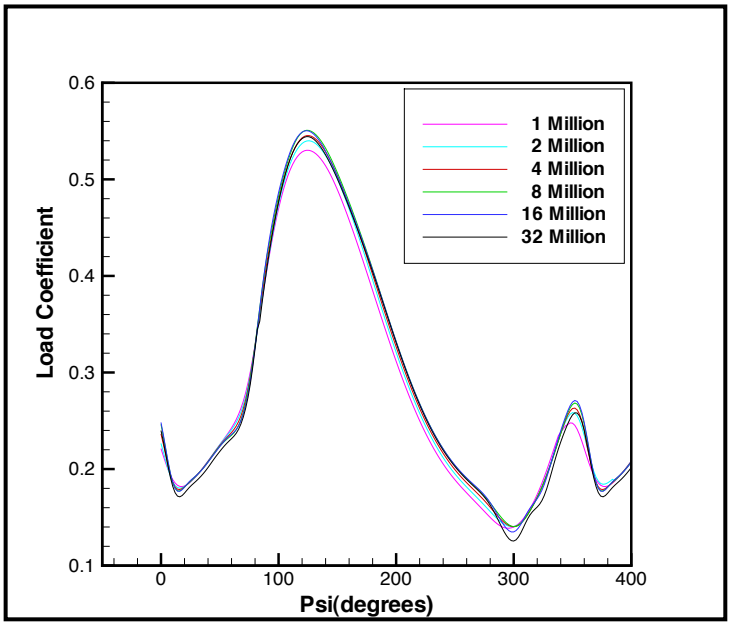
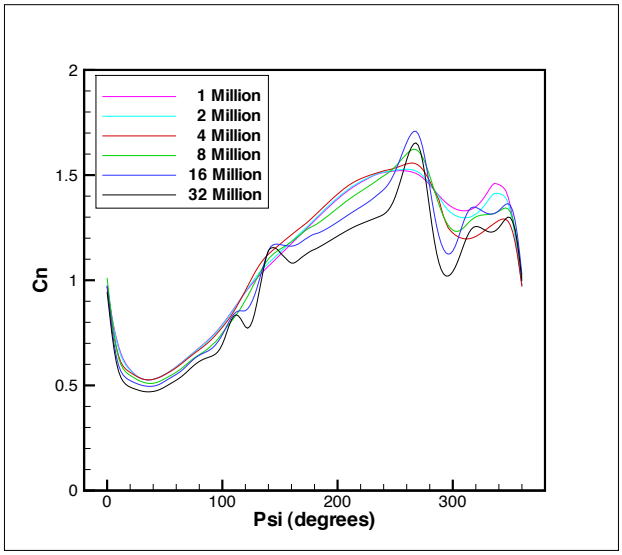


Figure 4. Lead blade total load coefficient around the azimuth for 1,2,4,8,16 and 32 million points. (Psi=0 corresponds to the blade pointing backwards, opposite to the flight direction).

data and multigrid data is computed and stored locally only. At any internal block boundary, ie connected to another block, the two adjacent planes of solution data are simply packed into an array and sent to the processor on which the neighbour block is stored, and vice-versa. Any incoming data is then unpacked according to an orientation flag. This allows all sends/receives to be non-blocking, and also makes future extensions straightforward, for example moving to a higher-order stencil.

To ensure maximum efficiency and, hence, grid sizes, there is no global data storage, so each processor only needs to store geometric and flow solution data for its blocks. This is possible due to the multiblock nature, which allows partitioning to be done separately. To this end a preprocessor has been developed, which scans the grid block sizes and maximises the load balance while also attempting to maintain the maximum number of multigrid levels.

Grid generation is also significant here. It is essential that the possible solution grid size is not limited by the grid generator and, hence, a multiblock generation tool has been developed that can generate a 1000 block, 64 million point rotor mesh in around 30 CPU minutes on a P4 Linux machine, requiring less than 2 GBytes RAM.



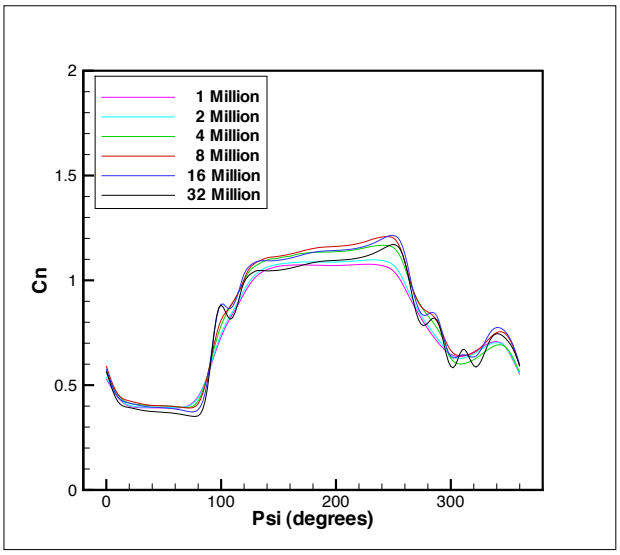


Figure 5. Sectional normal force coefficient (Cn) for blade section at 50% and 82% tip radius respectively, for 1, 2, 4, 8, 16 and 32 million points.

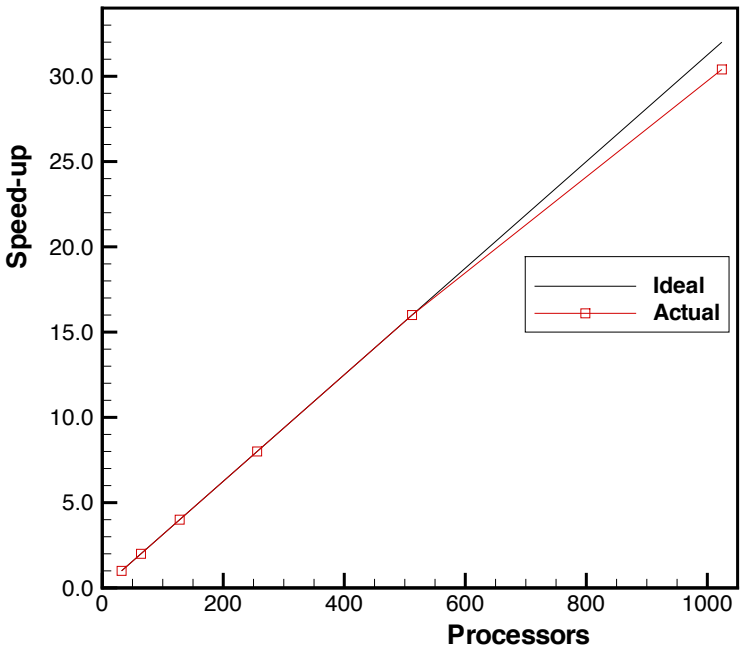


Figure 6. Scaling performance of code. Tests were performed on a 20 million point case.

Results and parallel performance

A wake grid dependence study was performed for a four-bladed rotor in forward flight. The computational domain, and block structure was kept constant and grids of size 1, 2, 4, 8, 16, and 32 million points were generated. Figure 1 shows the computational domain and block boundaries. The blade is the ONERA 7A rectangular blade, aspect ratio 15, ie tip radius is 15 blade chords, and the domain is a cylinder radius 50 chords, height 80 chords. There are 408 blocks. Figure 2 shows the grid in the rotor disk for 1, 8, and 32 million points.

The case chosen was a shallow descending case. The tip Mach number is 0.618, the advance ratio is 0.214 (so forward Mach number is 0.1322), and the rotor shaft is inclined 3.72 degrees backward. Hence, this has significant BVI effects. The results were computed on all six grids, using 180 real time-steps per revolution, and three revolutions were computed to obtain periodicity.

Figure 3 shows vorticity shading on selected grid planes, for the 1, 8, and 32 million point grids (the scales are the same in each). The effect of numerical dissipation is clear. The interesting consideration here is the grid density effect on local and global quantities. Figure 4 shows the total load coefficient of the lead blade around the azimuth. Hence, it is clear that if total load is of interest, then 32 million points are not required. However, figure 5 shows the local normal force coefficient, for blade sections at 50% and 82% of tip radius respectively, around the azimuth. The peaks are caused by BVI effects and, hence, it seems that grid convergence has not been achieved even with 32 million points.

As stated earlier, these calculations are extremely expensive. The 16 million point case was run on 256 CPUs on HPCx, and 32 million point case on 512 CPUs. The code has also been subjected to scaling tests, using a 20 million point mesh, and figure 6 shows the parallel performance (assuming a speed-up of 1 with 32 CPU's). Hence, excellent scaling has been achieved, and the code has been awarded a gold star for performance.

Acknowledgements

The HPCx computer time was provided through the UK Applied Aerodynamics Consortium (of which the author is a member) under EPSRC grant EP/S91130/01. Thanks are also due to Andy Sunderland, of the Terascaling Team, for performing the scaling tests.

Laser-driven double-ionisation: the HELIUM code on HPCx

B J S Doherty, J S Parker and K T Taylor, Department of Applied Mathematics and Theoretical Physics, The Queen's University of Belfast, UK.

Professor Ken Taylor is co-ordinator of the Multiphoton, Electron Collisions and Bose-Einstein Condensation HPC Consortium (e03). Barry Doherty is a Northern Ireland Department of Employment and Learning supported PhD student and Dr Jonathan Parker is an EPSRC supported PDRA.

Laser-driven helium is a scientifically crucial physical system which provides a first opportunity for the study of non-equilibrium two-electron quantum mechanical wavepackets. Knowledge and understanding of these multi-dimensional multi-electronic wavepackets and their properties is beginning to emerge through our ab-initio work, in close contact with complementary laboratory experiment on this system.

In moving from the Cray T3E architecture of the CSAR service to HPCx we considered it essential to re-engineer our HELIUM code so as to minimise inter-processor communication. We accomplished this over the period from October 2002 to March 2003 by changing from domain decomposition in angular momentum space to a configuration space domain decomposition. Moreover in this new decomposition we could, through exploiting a quantum mechanical symmetry, reduce, for any given problem, our memory and CPU demands by approximately 50%. We estimate overall that the re-engineered code is a factor of 5 more efficient on HPCx than the previous version.

With this re-engineered code, we have been able to make the world's first successful ab-initio calculation of double-ionisation of helium by high-intensity light from the most widely used laboratory short-pulse laser source, namely the Ti:sapphire laser operating at its fundamental wavelength of 780 nm. We are currently performing calculations of energy-resolved double-ionisation helium wavepacket spectra resulting from exposure of the atom to frequency-doubled Ti:sapphire light (ie 390 nm wavelength light). Such energy-resolved work for 780 nm is beyond the capability of phase 2 HPCx. This work is being carried out in close contact with world-leading experimentalists at Ohio State University in the USA making complementary measurements on helium at this laser wavelength.

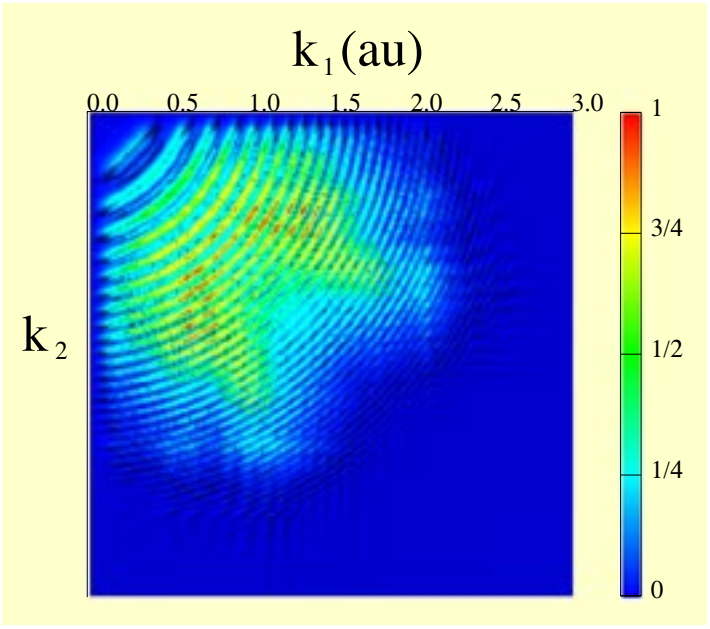


Figure 1. Probability density distribution of doubly-ionizing electrons in radial momentum space at the end of a 7-cycle 390 nm laser pulse of 9.0×10^{14} W/cm^2 peak intensity. The colour scale linearly maps probability density values from 10^{-4} (red) down to 0 (blue), with the colour legend in units of 10^{-4}.

The figure displays a typical energy-resolved two-electron ionisation wavepacket spectrum we have obtained, in this case for a laser peak intensity of 9.0×10^{14} W cm^{-2}. The figure was obtained through integration of the Schrödinger equation for the laser-driven 2-electron helium atom, a 5-dimensional time-dependent partial differential equation. Successful calculation of the physics of double-ionising wavepackets has required a numerical integration to 12 significant figures of accuracy. The colour scale runs from red indicating highest probabilities through yellow, green to blue at the lower probabilities. The axes plot magnitude of momentum carried by each electron, with the result that circular arcs in the figure occur at fixed kinetic energy shared between the two ionising electrons. A striking aspect of the figure is the succession of such circular arcs each separated in energy by the energy carried by a 390 nm photon. In this particular case it is possible to discern no less than 38 such rings. This very large number together with the strong probability density spread over individual rings indicates the highly non-perturbative strongly correlated character of the double ionisation process at this important laser wavelength. These complicated but fascinating dynamics are caused by three forces (of widely disparate geometries), ie laser-electron, electron-electron and electron-nucleus all coming into play to a comparable extent.

Given that twenty-five 390 nm photon absorptions are required to achieve initial ionisation of the 2 electrons, the figure represents a double-ionisation process involving at least 63 photon absorptions. Our closest competitors (in various countries around the world)

Continued opposite.

SC2004: bridging communities

Lorna Smith and Andrew Sunderland, HPCx Terascaling Team

The world's largest conference on high performance computing took place in Pittsburgh in November last year. Alongside 169 vendors including the usual industry heavyweights, over 100 universities, laboratories and other research groups were present. The conference is always a good place to find out about new and emerging technologies. One of the biggest splashes was made by IBM's new BlueGene L system, which topped the top 500 list with a peak performance 70 Tflops/s, displacing the Earth Simulator.

Field programmable gate arrays (FPGAs) also hit the exhibit floor with companies such as SGI and Cray demonstrating FPGAs closely integrated into HPC architectures, promising exceptional performance gains on critical components of an algorithm.

Edinburgh and Daresbury were busy at the event promoting HPC activities in the UK. Members of EPCC and Daresbury Laboratory (DL) jointly looked after the HPCx booth which featured highlights of the scientific and engineering-based research undertaken on HPCx. From the IBM stand Andrew Sunderland from DL gave a more detailed overview of scientific activities to the delegates with his presentation on Capability Science from the HPCx IBM p690+ Cluster in the UK.

EPCC presented a tutorial on performance scaling on constellation systems. It was particularly timely as constellation systems, or clustered symmetric multiprocessing (SMP) systems, have clearly become more prominent in the HPC market, most notably NASA's new SGI Altix 3700 cluster which clocks in at an impressive 51.87 Tflop/s. We focused on the tools and techniques required to achieve optimal performance and scaling on these systems, looking at topics such as optimising inter- and intra-

node communication, such as overlapping communication, cluster aware message passing, mixed mode programming and processor mapping.

EPCC's Neil Chue Hong was selected to make a presentation in the HPC Software Challenge, which honours participants working to improve the productivity of HPC software developers and the quality of HPC software. Neil talked about OGSA-DAI, a highly successful UK project involving Edinburgh, Manchester and Newcastle universities, with industrial participation by IBM and Oracle. The project has developed middleware to assist with access and integration of data from separate data sources via the grid.

Other highlights included SC Global 2004, the Access Grid-enabled component of SC2004, which included a first-ever demonstration of a simultaneous connection of AG nodes from all six inhabited continents. All in all a successful event was had by all, with a good UK presence. Seattle beckons for 2005.

Laser-driven double-ionisation

Continued from opposite page.

are capable of calculating no higher than two-photon absorption double-ionisation processes. They are consequently limited to weak-field calculations for laser wavelengths of the order of 30 nm which are not yet experimentally feasible. Current developments in Free Electron Lasers (FELs), however, should make experiments involving few-photon double ionisation of atoms possible shortly.

The physics of 2-electron atoms in intense laser radiation is currently a hot topic world-wide, with the experimental study typically performed at the laser wavelengths we alone can seriously address with ab-initio theory, ie with the ubiquitous Ti: sapphire laser at a wavelength of 780 nm or frequency doubled Ti: sapphire at a wavelength of 390 nm. The computational difficulty encountered in the theoretical analysis scales as the wavelength cubed. Integration of Schrödinger's equation to obtain energy-resolved doubly-ionising wavepackets for a laser wavelength of 390 nm, for instance, is $13^3 \approx 2000$ times more computationally demanding than the corresponding 30 nm calculations performed by our competitors. Similarly 780 nm is $2^3 = 8$ times more demanding than 390 nm.

The present energy-resolved calculations at 390 nm require a minimum of 861 HPCx processors, each using its maximum user-available memory. The calculation represented in the figure required 50 wall-clock hours on 861 HPCx processors. Future calculations for the corresponding energy-resolved response to 780 nm laser light will require the re-engineered HELIUM code exploiting the full capability of an upcoming HECToR service.

Contact Professor Ken Taylor for further details:
k.taylor@qub.ac.uk

Forthcoming events

International Conference on Parallel Computing (ICPP 2005), Oslo, 14-17 June 2005: http://www2.dnd.no/icpp2005

20th International Supercomputer Conference, 21-24 June 2005, Heidelberg: http://www.supercomp.de

Euro-Par 2005, 30 August- 2 September 2005, Lisbon: http://europar05.di.fct.unl.pt

Control and Prediction of the Organic Solid State Open Day, University College London, 13 September 2005: http://www.cposs.org.uk

12th International Parallel Virtual Machine and Message Passing Interface Conference (EURO PVM MPI 2005), Sorrento, 18-21 September 200: http://www.pvmmpi05.unina2.it

Supercomputing 2005: Gateway to Discovery, 12-18 November 2005, Seattle, http://sc05.supercomputing.org. This conference will include a trans-continental Grid computing demonstration linking HPCx, CSAR and HPC services in the USA, as a follow-up to the TeraGyroid project described in *Capability Computing* issue 2: http://www.hpcx.ac.uk/about/newsletter (see also: http://www.cse.clrc.ac.uk/about_us/Frontiers2004).

3rd HPCx Annual Seminar, 5 December 2005 and Daresbury 16th. Machine Evaluation Workshop, 6-8 December 2005, Daresbury Laboratory, Cheshire (provisional dates: http://www.hpcx.ac.uk and http://www.cse.clrc.ac.uk will provide confirmation and details in the near future).

SciComp12, Spring 2006, NCAR, Colorado: http://www.spscicomp.org.

MSc in High Performance Computing

EPCC at the University of Edinburgh offers a one-year taught Masters course: the MSc in High Performance Computing (HPC). This well-established programme provides an excellent grounding in HPC technologies and their practical application.

The MSc in HPC will appeal to students who have a keen interest in programming and would like to learn about HPC and parallel computing. We aim to attract students who hold degrees in the physical sciences, engineering, computer science or mathematics, or who have equivalent work experience. The course has a strong practical focus and students will have access to world-leading HPC platforms and technologies.

Applications are welcome from graduates of all areas of science, engineering, computer science and mathematics, and from those currently working in a relevant field. The entrance requirement is a good honours degree or equivalent work experience. No prior HPC knowledge is assumed, but candidates must be competent programmers in Java, C++, C or Fortran. A number of funded studentships are available for UK and EU residents. It is also possible to study part-time: please contact the MSc programme coordinator for more information.

The MSc is supported by EPSRC, who offer a number of studentships.

Taught Courses

- Fundamental Concepts of HPC
- Object Oriented Programming for HPC
- Practical Software Development
- Hardware, Compilers and Performance Optimisation
- Shared Memory Programming
- Applied Numerical Algorithms
- Message Passing Programming
- Exploiting the Computational Grid
- Parallel Decomposition
- Scientific Visualisation
- Core Topics in HPC and e-Science
- Advanced Topics in HPC and e-Science

About us

The HPCx system is located at the UK's CCLRC Daresbury Laboratory and operated by the HPCx Consortium (UoE HPCx Ltd).

HPCx is led by the University of Edinburgh, with the Council for the Central Laboratory for the Research Councils (CCLRC) and IBM. The project is funded by EPSRC.

Capability Computing 5 was edited and put together by Martin Plummer and Tracy Peet.

HPCx Technical Reports:
http://www/hpcx.ac.uk/research/hpc

HPCx website:
http://www.hpcx.ac.uk
Helpdesk: support@hpcx.ac.uk
Telephone: 0131 650 5029
Fax: 0131 650 5029

UoE HPCx Ltd,
c/o EPCC,
University of Edinburgh,
JCMB, Mayfield Road,
Edinburgh, EH9 3JZ