In this issue...

2 SPRINT: helping biostatisticians exploit HPC systems
3 HECToR goes quadcore
4 CP2K: High Performance Quantum Chemistry on HECToR
5 The MSc in HPC: Another year over, a new year begins
6 HPC-EUROPA visitor programme at EPCC
7 Our Research website goes live
   Undergraduate teaching at EPCC
8 Maxwell and the PRACE project: Accelerating to the future
9 EUFORIA reaches the halfway mark
10 QCDOC: 5 years on
12 Multicore training event report
Welcome to the sixty-sixth EPCC News. It’s a busy time at the centre with many diverse activities happening, from machine upgrades to participation in Europe-wide projects to preparing for another academic year of teaching. I hope this issue will give you an interesting overview of what’s going on, as it has given me during my first time as editor.

Continuing the multicore theme from the last issue, the upgrade of HECToR from dual core to quad core processors has been completed successfully. Liz Sim gives the details of this "Phase 2a" upgrade. On a smaller scale, the trend towards multicore everywhere continues – as parallel computing goes more and more mainstream, it is becoming increasingly beneficial for students of many different disciplines to be able to understand and make use of it. Kevin Stratford talks about EPCC’s contribution to undergraduate teaching at the University, while David Henty reviews the new multicore programming course and Judy Hardy looks at EPCC’s successful MSc in High Performance Computing, now entering its ninth year.

Indeed, the range of projects currently running in the centre and featured in this issue illustrates that parallel and novel computing applies to a broad range of sciences: gene analysis with SPRINT, quantum chemistry with CP2K, nuclear fusion research and development with EUFORIA, and more.

Looking to the future and building on our previous work in this area, EPCC is investigating the performance and programmability of FPGA accelerators as part of the Europe-wide PRACE project. Paul Graham looks at the potential of this novel technology and some of the challenges ahead.

It’s an exciting time to be in HPC. I hope you enjoy this issue.

Wellcome Trust grants further funding for SPRINT

Developing software that allows life scientists to concentrate on research problems rather than computer problems.

Gene analysis is becoming increasingly complex and can be greatly enhanced by exploiting the power of high-performance computing (HPC), but the software can be difficult for researchers to use. To allow greater access to the benefits of HPC, EPCC and the Division of Pathway Medicine (DPM) at the University of Edinburgh developed a prototype framework called SPRINT (Simple Parallel R INTerface), which allows biostatisticians to more easily exploit HPC systems. The Wellcome Trust has now funded the SPRINT project for a further two years. This will allow the development of the SPRINT framework and for a number of commonly used functions to be added to use its ability by a wider community. SPRINT is an easy-to-use parallel version of R, a statistical language that processes the data gleaned from microarray analysis, a technique which allows the simultaneous measurement of thousands to millions of genes or sequences across tens to thousands of different samples.

Processing the data that is produced by microarray analysis tests the limits of existing bioinformatics computing infrastructure. A solution is to use HPC systems, which offer more processors and memory than desktop computer systems. However, R must be able to utilise multiple processors if it is to fully exploit the power of HPC systems to analyse genomic data. There are existing modules that enable R to do this, but they are either difficult for HPC novices or cannot be used to solve certain classes of problem. SPRINT allows parallelised functions to be added to R without the need to master parallel programming methods, enabling the easy exploitation of HPC systems.

Prof. Peter Ghazal, director of the Division of Pathway Medicine, said, “SPRINT will greatly increase the computing power available to many researchers and is therefore a unique opportunity to accelerate the discovery of the genes linked to diseases.

"SPRINT requires very little modification to existing sequential R scripts. As an example the project team created a function that carries out the computation of a pairwise calculated correlation matrix. This performs well with SPRINT. When executed using SPRINT on an HPC resource of eight processors this computation reduces by more than three times the time R takes to complete it on one processor."

Recently the joint EPCC/DPM SPRINT team was awarded a grant for computational science and engineering support to optimise SPRINT for use on HECToR, the UK’s national supercomputing service. This grant, administered by NAG Ltd, will enable even larger microarray analyses to be performed by allowing SPRINT to take advantage of the multicore capabilities of today’s supercomputers.

www.r-sprint.org
HECToR: One great step on the multi-core ladder

Liz Sim

July 20th 1969, Neil Armstrong takes his first great step on the moon. 40 years later, on July 19th, the UK’s high-end computing resource takes another great step on the multicore ladder.

July 19th saw the HECToR service move from dual-core to quad-core processors. The upgrade, known as the ‘Phase2A upgrade’ was funded by the UK Research Councils and was staged over a two week period. During this time all 5664 nodes were converted to quad-core processors. This amounts to a total of 22,656 cores, each of which acts as a single CPU. The processor used is an AMD 2.3 GHz Opteron. Users can now run simulations using up to 16,384 processors on HECToR.

During the upgrade, all 22,656 DIMMs were also upgraded. Each quad-core socket now shares 8GB of memory, giving a total of 45.3 TB over the whole XT4 system. Thanks must go to the team of dedicated engineers from CRAY who spent many hours onsite at the University of Edinburgh’s Advanced Computing Facility (ACF), upgrading the HECToR hardware. The team comprising of local CRAY staff, and engineers drafted in from both the US and other European CRAY sites had planned well and were highly organised. In total 28,320 components were swapped.

The HECToR upgrade was split into two phases to minimise the disruption to users. Half of the service was available for use whilst the other half was being upgraded. The hardware upgrade and testing went to plan with no major issues encountered. Preventative measures were taken by the team to ensure that the upgrade did not impact the live service. This planning was worthwhile as there was no impact on the main service as a result of the upgrade work. The system successfully completed the acceptance tests phase in early August.

This upgrade increased the theoretical peak performance of the service from 59TFlops to 208TFlops. Unfortunately, the timing of the upgrade meant that we just missed the deadline for the June edition of the Top500. A test Linpack run demonstrated a maximum performance of 174 TFlops. This would have placed HECToR at a theoretical sixteenth place in the list. We will have to wait with bated breath to see where we are placed in November when the next list is unveiled at SC09. If you are attending SC09 please come and visit EPCC at booth 658. We would love to hear your comments and questions about the HECToR service.

Planning has already started on the next phase of the HECToR upgrade – Phase2B. In March 2010, a 20-cabinet CRAY ‘Baker’ system will be available. This is a major step on the multi-core ladder as the Baker will be using 12-core processors. The system will consist of 44,544 cores in total, delivering an estimated peak performance of over 300 TFlops. In addition to the ‘Baker’ system, approximately half of the existing XT4 system will be retained. This will be followed by an upgrade to the network. In late 2010 we will move to the Gemini interconnect.

Following on from the Phase2A upgrade, we have redesigned the HECToR website. If you are new to HPC and have tried to investigate what HECToR has to offer, you may have found the original site confusing as there was a lot of in-depth information for existing users. We have now introduced a new public facing webpage, with simple facts and examples of what HECToR can do for you. There are case studies from existing users, basic facts on the service available, and simplified guidelines on how to apply for access.

One small step today could lead to one giant leap in your research...

To find out more about the HECToR service, including how to apply and the cost of access for both the XT4 and the X2 vector system, see: www.hector.ac.uk
In the Summer 2008 edition of EPCC News, we reported that EPCC had begun a collaboration with the CP2K development group at the University of Zurich, to improve the performance of CP2K on HECToR. An initial year-long project of optimisation of key areas of this code has now been completed with great success.

CP2K [1] is a freely available, open source application which uses Density Functional Theory (DFT) to perform ab-initio quantum mechanical calculations on a variety of physical systems. It can be used to predict highly accurate molecular and crystalline structures and properties, and determine the evolution of systems using molecular dynamics, amongst many other features. Since its release in 2000 CP2K usage has grown dramatically and this is expected to continue with the recent addition of support for hybrid functionals for DFT. CP2K is heavily used on HECToR – Dr Ben Slater (co-propoer of the project) and his research group at University College London are using CP2K to study the properties of exotic phases of water ice. Dr Carole Morrison, University of Edinburgh, uses CP2K to study proton transport in biological systems. In all, around 50,000 CPU hours are used each month on HECToR by CP2K.

Iain Bethune, an Applications Consultant at EPCC, undertook detailed profiling and analysis of the code from August 2008 to July 2009. Several optimisations and improvements were made, including restructuring communications in the key ‘realspace to planewave’ transformation which is central to CP2K’s implementation of the Gaussian and Plane Waves (GPW) method. The computationally expensive parallel 3D Fourier transform was also modified to reduce parallelisation overheads, and to allow the used of advanced performance planning of the Fastest Fourier Transform in the West (FFTW) library [2]. As described earlier, CP2K can be used to study nonhomogeneous systems including biological molecules, and solid/liquid/gas boundaries, which exhibit poor load balancing on massively parallel supercomputers such as HECToR. Modifications were made to the existing load balancing algorithm to allow a better spread of work across all available processors. These improvements were contributed back into the CP2K CVS repository immediately, after thorough testing with an automated suite of regression tests, which ensure the continued accuracy of the code.

At the end of the project, substantial performance improvements had been achieved – up to 30% on 256 cores for a small benchmark system of liquid water, and up to 300% on 1024 cores for a larger, non-homogenous system. These improvements are already available to users of HECToR and HPCx via the centrally installed versions of CP2K, and are available to CP2K users worldwide through the CVS repository on the CP2K website.

Following on from this project, another dCSE grant has been obtained to improve the scalability of the code further using a hybrid MPI/OpenMP approach. This will allow CP2K to take greater advantage of the increasingly multi-core nature of modern supercomputers, such as the recently installed quad-core AMD Opteron processors of HECToR.

This work was funded under a HECToR Distributed Computational Science and Engineering (dCSE) grant from NAG Ltd. We are grateful for the in-depth technical support and direction provided by Dr Joost VandeVondele (University of Zurich), who contributed greatly to the success of the project.

The dissertation project forms a very important – and enjoyable – part of the MSc. After completing their taught courses, students have the opportunity to work over the summer on a 16-week individual research project. A wide variety of topics is available, reflecting the extensive range of staff skills and interests at EPCC. Here we highlight four dissertations from 2008/09. Past dissertation reports are also available on the MSc website.

‘GPU Acceleration of HPC Applications’
Alan Richardson
The use of Graphics Processing Units for non-graphical computations (GPGPU) has generated a large amount of interest due to the promise of impressive performance gains. The process of accelerating existing HPC applications through the use of GPUs is investigated in this dissertation. Five candidate codes were considered – CENTORI optimisation library, FAMOUS, Ludwig, CHILD, and CSMP – of which three were ported to partially run on a GPU using Nvidia’s CUDA language.

‘Performance of the network in a Cray XT4 machine: Hot spot analysis and optimization of the communications’
Pablo Barrio
This dissertation is a general study of the network in HECToR. Its main component is a Cray XT4 system, with a very similar network to that found in the newer XT5 machines. This dissertation includes a general benchmarking of the network, followed by a more specific testing with micro benchmarks. These stages have shown specific situations where the timings are affected by the network operation. Real codes have also been used and a mechanism to optimise task placement has been tested. The data shows that optimization is possible by rearranging computing tasks into physical if the work decomposition inside the code is known in advance.

‘Message Passing Library for Java’
Rajesh Babu
Message Passing Interface (MPI) is the standard message passing API for parallel computing. The original standard defines bindings for C, FORTRAN and C++. Several research efforts have extended the MPI standard for Java. But none of them really performed well enough compared to standard MPI implementations. In this project we designed a new message passing library for Java, called MPLJava, taking some of the best practices followed in the existing libraries and improvising them with the latest improvements in Java technology. We discuss the architecture of our implementation, its performance compared to standard MPI implementations, and propose some future research direction.

‘UPC Collective Optimization’
Savvas Petrou
Efficient collective operations are an important feature for new parallel programming languages. The size of massively parallel machines is increasing fast and the performance of these operations is crucial to the scalability of many applications. The Unified Parallel C (UPC) language is one of the new parallel programming languages based on the Partitioned Global Address Space (PGAS) model. The purpose of this project is to investigate the performance and scaling of the current implementations of collective operations and also develop new implementations based on tree structure algorithms.

http://www.epcc.ed.ac.uk/msc
Many readers will know that EPCC plays a leading role in HPC-Europa2, the EC funded inter-disciplinary research visit programme, which supports scientists in gaining access to HPC facilities at either EPCC or one of our partner centres, while working in collaboration with a local research group working in a similar field.

UK-based researchers can benefit from HPC-Europa2 either by visiting another European research group, or by hosting a researcher within their own group.

One recent visitor to EPCC was Dr Monika Pietrzyk, of the Non-linear Optics Group at the Faculty of Physics at Warsaw University of Technology. Dr Pietrzyk spent nearly 3 months in Edinburgh, working with Professor Derryck Reid, head of the Ultra-fast Optics Group at the Department of Physics at Heriot-Watt University, on her project "Propagation of few-cycle optical pulses in nonlinear guiding structures".

Dr Pietrzyk chose to visit this research group in order to compare the results of her own numerical simulations with experimental research on ultrashort pulses carried out by the group at Heriot-Watt.

Due to limitations of the multisymplectic integrator underlying the Warsaw group’s previous work on pulse compression and shock formation, access to more powerful computing facilities had become necessary in order to carry out further numerical simulations. These simulations and their analysis should lead to a better understanding of the properties of few-cycle pulses, and could be used to develop non-linear photonics applications in the future.

During the course of her visit, Dr Pietrzyk optimised and parallelised her Fortran90 code using MPI, with the help of the HPC-Europa2 team at EPCC. Her calculations involve propagating and evolving the wavefunction of a short pulse, and although non-trivial, it is a problem well suited to parallelisation. Dr Pietrzyk also developed a new set of techniques in conjunction with Professor Reid and other hosts at Heriot-Watt University, and embarked on an implementation of these during her visit.

Professor Reid rated the scientific value of the work performed during this visit as excellent, commenting that, “The visit created a potentially very useful link between my group and a talented theorist”.

Closing dates for HPC-Europa2 applications are held 4 times per year. In 2010 these are expected to be: 28th February, 15th May, 31st August and 15th November. For further information, including the on-line application form, see: www.hpc-europa.eu/
As part of the School of Physics at The University of Edinburgh, EPCC contributes several courses to the undergraduate teaching programme. The undergraduate student in Scotland is typically offered a 4-year bachelors degree – in this case a BSc Honours in Physics – giving a broad view of the subject and a good basis for many prospective careers. Edinburgh also offers a 5-year Master of Physics (MPhys) degree aimed at those intending to take up the science as a profession.

EPCC brings to bear skills in both software engineering and scientific computing which are both important for the general education of those interested in entering today’s job market. Introductory courses on computer simulation using Java are popular, while at the other end of the spectrum courses in more advanced models including DNA unzipping and fluid dynamics. In addition, the introduction of multi-core laptops and desktops means that many students can now see the advantages of parallel algorithms in their own work at home, rather than regarding parallelism as an esoteric and unreachable activity requiring specialist equipment.

Going live...

EPCC’s new online Research Collaboration pages

EPCC is committed to making the power of advanced computing available to all areas of research. Our staff work on collaborative projects across a wide range of disciplines, and are enthusiastic about forming new research relationships and taking on new technical challenges.

Please take some time to browse our new-look Research Collaboration web pages, which describe our current activities and give details on how to get involved in collaborating with EPCC.

Visit the new pages here: www.epcc.ed.ac.uk/research-collaborations

Undergraduate teaching at EPCC

Kevin Stratford

As part of the School of Physics at The University of Edinburgh, EPCC contributes several courses to the undergraduate teaching programme. The undergraduate student in Scotland is typically offered a 4-year bachelors degree – in this case a BSc Honours in Physics – giving a broad view of the subject and a good basis for many prospective careers. Edinburgh also offers a 5-year Master of Physics (MPhys) degree aimed at those intending to take up the science as a profession.

EPCC brings to bear skills in both software engineering and scientific computing which are both important for the general education of those interested in entering today’s job market.
PRACE (Partnership for Advanced Computing in Europe) is intended to prepare for the creation of a persistent pan-European HPC service, based on an infrastructure of around 4–5 Petaflop-scale supercomputers. There are PRACE partners in most European countries who work on activities ranging from the governance and legal framework for the future infrastructure, through dissemination and training, to a number of technical activities looking at possible hardware and software for Petascale systems.

One of EPCC’s roles in the project is contributing to workpackage 8: “Future Petaflops/s computer technologies beyond 2010”. Amongst other things, this workpackage is assessing and evaluating emerging HPC technology in order to inform the decision making process when the next generation of multi-petascale HPC systems are being procured. Within PRACE WP8 four compute kernels have been identified which represent typical algorithms encountered in HPC: a random number generator (RNG), a matrix by matrix multiplication (MM), a sparse-matrix by vector multiplication (MV), and a Fast Fourier Transform (FFT). These kernels are being used as benchmarks to evaluate various existing and emerging compute architectures such as GPUs and Cell processors. As part of the evaluation process it is not just base performance that WP8 is interested in: aspects such as programmability, and power requirements versus performance (“Flops per Watt”), are also being considered. EPCC is looking at Field Programmable Gate Array (FPGA) technology in this context.

Since 2007, via the FPGA High Performance Computing Alliance, EPCC has had “Maxwell” – a demonstrator system comprising of 64 FPGA accelerator cards (32 from Alpha Data Ltd., 32 from Nallatech Ltd.). These cards are hosted in a 32 node BladeCenter system, each node therefore consisting of a single-core 2.8GHz Xeon processor with two accelerator cards connected on a PCIX-bus. The main novelty in the design of the system is its use of the RocketIO connections on the FPGAs to create a nearest neighbour two-dimensional mesh interconnect between the FPGAs. This can allow a complete numerical algorithm to be executed in parallel with no need for the FPGAs to communicate over the (relatively) slow PCIX-bus. However, within the PRACE project we are confined to looking at single-FPGA execution.

In order to better understand the challenges of programming for FPGAs, EPCC has decided to take a two-fold approach to implementing the kernels on the accelerators. Firstly, a VHDL (a relatively low-level language typically used for programming FPGAs) version of a kernel will be developed – the expectation being that this, properly tuned, will give the best performance rate. Secondly, a “C-to-gates” compiler would be used – these tools allow code written in a subset of C to be compiled to VHDL. These compilers make FPGA programming much more accessible to the general programmer, but historically rarely attain the performance obtained by hand coding VHDL. This method will allow a comparison of programmability and performance from the two different approaches. The C-to-gates tool we have chosen is the Harvest Compile Environment (HCE) from Ylichron: this boasts support for a large subset of C.

As of August 2009, two of the kernels have been tackled: the random number generator and the matrix by matrix multiplier. The initial results are quite interesting: for the RNG, the VHDL versions of the kernel can produce in the order of 900 million values per second, as compared to 14.4 million per second on the host Xeon (about 60x more values per second on the FPGA). However, due to the hardware restrictions of Maxwell (the connecting PCI bus is relatively low-bandwidth), the values can only be taken off the FPGA at a rate of four million a second, seriously lowering the apparent performance. However, one can imagine that in a real HPC program, the values would be used by another part of the algorithm, and thus may be consumed on the FPGA or by another FPGA via RocketIO, so transport across the PCI bus could be avoided. For the VHDL version of the matrix by matrix multiplier, speedups of 3x on the FPGA, and 2x including transporting data back across the PCI bus have been achieved, as compared to the host performance.

Initial results with the C-to-gates compiler have been less promising, with the generated VHDL code running slower than the host for both kernels. However, there has so far only been a small amount of effort put into investigating this: EPCC are working with Harvest to understand the best approach to coding the original C in order to assist the compiler in optimisation, which is expected to improve the performance significantly.

The remaining plan is to further investigate the C-to-gates tool and how best to get performance from its generated code, and to then tackle the two remaining kernels. The output from this work will be integrated with similar investigations of other architectures from the PRACE partners, and will appear in a public deliverable from the PRACE project due in December 2009.

Contact Paul Graham for more information:
pgraham@epcc.ed.ac.uk
EUFORIA (EU fusion for ITER Applications), a European funded project which aims to provide HPC and Grid resources and expertise to the European fusion research community, has reached its halfway point. EUFORIA is a collaborative project which involves 13 European institutions, including EPCC, working in partnership with the fusion community. It does not directly fund fusion scientists, but relies on their co-operation and involvement in providing their simulation codes for optimisation and porting.

EPCC is involved in a range of activities in EUFORIA, focusing mainly on the porting, parallelisation and optimisation of fusion simulation codes for HPC machines. This includes providing access and resources on HECToR for fusion scientists with over one million AUs already distributed. However, we are also heavily involved in providing training and documentation for the European fusion community, and running the user support helpdesk which aims to make users’ experiences of using Grid and HPC resources as painless as possible. The very fact that user support, with an explicit user helpdesk, is part of a project like this is encouraging as all too often user support is forgotten in the pursuit of scientific research and resource provision.

The direction of the EUFORIA project has coalesced around a workflow engine, Kepler, which European fusion scientists (co-ordinated by the Integrated Tokamak Modelling Task Force[1] or ITM) have chosen as their tool for scientific simulation. The goal of the ITM is to develop a numerical tokamak, a computational model of a fusion reactor. However, this is a very challenging task as it involves bridging the time and space scales separating turbulence simulations and a whole fusion reactor. A typical turbulence flux-tube calculation might encompass a few centimetres of radial extent compared to the meters of next generation fusion reactors (like ITER[2]); between a hundred thousand and one million time steps are typically required for a fusion reaction to reach a steady-state. This means there is a very high computational requirement to model even parts of a full fusion tokamak, and any attempt will involve undertaking a number of different simulations.

A scientific workflow demonstrates one way of addressing this problem: a number of flux tubes are used to calculate the flux of particles and heat at different radial locations across the plasma; these fluxes are then used by a transport code to update the density and temperature profiles; these updated profiles are again used by the turbulence codes. Each of these steps is undertaken by separate simulation codes which have distinct computational requirements. For instance, a plasma turbulence or equilibrium code such as GEM or HELENA will require hundreds to thousands of processors for tens of minutes for an average run. A typical transport code, which links the output of turbulence and equilibrium codes, currently requires a small number of processor for a few seconds. Using the workflow engine Kepler the ultimate goal of EUFORIA is to provide users with the functionality to run fusion workflows using components that exploit both HPC resources and the EUFORIA Grid (depending upon their computational requirements) and visualise data generated either from the running workflow (to allow computational steering) or once the workflow has finished. Currently EUFORIA has demonstrated the feasibility of running complex workflows on a single HPC machine and extended Kepler to allow it to run jobs on either the EUFORIA Grid or HPC machines.

Fusion is not alone in pursuing workflow engines or other technologies (such as portals) to simplify the process for scientist to exploit computational resources (whether they be Grid or HPC or both). There is a proliferation of differing workflow engines, portal technologies, and Grid abstraction layers that are being developed and used by different scientific communities. Which ones survive the test of time to become established and widely used still remains to be seen.

EUFORIA: http://www.euforia-project.eu

References
The fourteen-thousand node QCDOC machine arrived at the ACF in Edinburgh in November 2004 and, once the initial shake-out phase was complete, has been running almost constantly ever since. Twelve cabinets are shown in figure 1. This bespoke computer has enabled the UKQCD collaboration [1] to do calculations to understand the properties of quarks which otherwise would have not been possible without building their own computer.

Quarks are the fundamental particles making up 99.9 per cent of ordinary matter. They are bound together by the strong nuclear force, mediated by the exchange of gluons. The theory of quark and gluon interactions is Quantum Chromodynamics, or QCD. Supercomputer simulations enable scientists to ‘look inside’ quark and gluon bound states, such as the proton and a plethora of other states known collectively as hadrons. The calculations are performed by constructing a discrete four dimensional space-time grid (the lattice) and then solving the fundamental QCD equations on this grid. Such lattice QCD simulations are the only known first-principles method for studying hadronic interactions.

Due to the local nature of the interaction, lattice QCD can be domain decomposed where the communication pattern is nearest neighbour only and is thus ideally suited to the massively parallel processing paradigm. The amount of work each processor has to do depends on local volume. The smaller this is, the greater the proportion of data is ‘near’ the processor. Thus each processor has less work and can access the data faster. However, the smaller the volume, the greater the number of communications of a smaller size must be performed. So the performance of lattice QCD calculations are governed by the following machine characteristics: memory bandwidth, and the latency and bandwidth of communication.

QCD-on-a-chip (QCDOC) was designed and built by a collaboration of researchers from the Universities of Columbia and Edinburgh, the Riken-Brookhaven Research Center, IBM T.J. Watson Research Laboratory using the IBM system-on-a-chip technology [2]. The design target was a processor with high bandwidth but small on-chip memory and a low latency, high bandwidth nearest neighbour communications network. The processor of the QCDOC ASIC is an IBM PPC 440 with a 64-bit floating point unit capable of one multiply and one add per cycle. There are two main custom design components. The first is the memory access system, which has eight Gbyte/s...
bandwidth to four MByte embedded on-chip DRAM. The second, is the Serial communication unit (SCU). This manages the twenty four uni-directional links arranged in six-dimensional torus to the nearest neighbours. The SCU has a DMA engine which can place data directly into the local memory of the receiving processor. Two ASICs are placed on a daughter card, 32 daughter boards sit in a motherboard, eight motherboards are arranged in a crate and two crates are placed in a rack of 1024 nodes.

Shown in figure 2 are some performance figures for the code on different machines. The QCDOC machine sustains a much higher percentage of peak performance especially at small local volumes, than either the Cray XT4 (HECToR), or a Blue Gene/L. All of these machines are housed at the ACF in Edinburgh. The QCDOC machine was designed to run with small local volumes. Total performance is achieved scaling to large numbers of processors, typically 4096.

QCDOC has been remarkably reliable. Almost all of the original fourteen thousand processors are still functional. The UKQCD collaboration has been able to run QCDOC almost continuously. In terms of scientific output QCDOC has been a great success. With QCDOC it was possible to use a new five-dimensional formulation of QCD, Domain Wall Fermions (DWF), with better symmetry properties. This meant that systematic uncertainties in the calculation were significantly reduced. The DWF project, a joint collaboration between RBC [3] and UKQCD collaborations, has produced around 20 papers including three Physical Review Letters [4]. This new formulation run on QCDOC has enabled simulations to be performed at sufficiently light quark masses to provide a significant test of the Standard Model once compared to experiment.

QCDOC is still running, but due to STFC funding delays it has moved from a national facility to a University of Edinburgh facility, running a reduced service of 4096 nodes. It is hoped to continue this until funding for Blue Gene/Q can be secured.

References
[1] ukqcd.epcc.ed.ac.uk

Above: Figure 2 shows some performance figures for the code on different machines.

Right: The Edinburgh plot shows the self-consistency of the simulations run on QCDOC. Shown are the ratios of the masses of the pion (the lightest state in QCD, labelled 'P'), the rho meson (labelled 'V') and the proton, for different quark masses, lattice spacings and volumes. All data generated by the Domain Wall Fermion project on QCDOC machines in Edinburgh and Brookhaven National Lab, USA. (Cite C. Allton et al, Phys.Rev.D78:114509,2008, arXiv:0804.0473v1).
Report: Multicore Training Event
8–10 September 2009, Edinburgh

As advertised in a previous edition, EPCC ran a three-day multicore workshop “Exploiting Multicore Processors: Challenges and Programming Models” at the e-Science Institute in Edinburgh at the beginning of September. The aim was to introduce the basic concepts of multicore hardware and software, and to illustrate them with a range of practical exercises using the OpenMP language extensions for C, C++ and Fortran.

The workshop was very well attended with some 30 people coming from a range of academic and industrial backgrounds. Although EPCC has run courses in this area for many years, currently as part of the MSc in HPC, we have previously targeted specialist parallel supercomputers. However, with multicore CPUs now commonplace on the desktop, we widened the scope for this workshop and encouraged attendees to run the exercises on their own laptops if possible.

We were pleasantly surprised to find that almost everyone was able to run successfully on their personal machine with only a modest amount of tweaking. The major issue was, as is often the case, the amount of stack space available at runtime. However, this was usually fixed with a combination of additional linker options or ulimit settings. Luckily, we did always have the backup of EPCC’s faithful 16-core "ness" system!

Participants developed and ran OpenMP codes on both dual-core and hyper-threading hardware, and under a variety of Linux, Mac and Windows operating systems. In comparison to performance on high-end machines, we noticed that memory bandwidth was even more of a limiting factor. Interestingly, many people also observed improved performance as the number of threads was increased above the number of physical cores. This appeared to be due to the fact that a fully-fledged desktop OS has many more tasks to perform than a stripped-down OS on a dedicated supercomputer. Running more threads means that the user gets a greater share of the total CPU resources by sheer weight of numbers! However, the overheads of additional threads soon outweigh this gain and performance starts to degrade. This would suggest that users might benefit from increasing the scheduling priority of their own jobs even though they are the sole person using their machine.

Feedback from the course has been excellent with an average overall score of 89% from some 20 anonymous forms, so we hope to be able to run something similar again. One clear message from attendees was that they would appreciate more information on multicore-aware debugging and profiling tools, so this is something we aim to address in the future.

This course was funded by EPCC with additional support from the e-Science Institute, the University of Edinburgh’s Roberts Fund and registration fees from commercial attendees.