

HPCx: A NEW RESOURCE FOR UK COMPUTATIONAL SCIENCE

Mike Ashworth, Ian J. Bush, Martyn F. Guest, Martin Plummer and Andrew G. Sunderland
*Computational Science and Engineering Department, CLRC Daresbury Laboratory, Daresbury,
Warrington, Cheshire, WA4 4AD, UK*
Email: m.ashworth@dl.ac.uk, m.f.guest@dl.ac.uk

Stephen Booth, David S. Henty, Lorna Smith and Kevin Stratford
*Edinburgh Parallel Computing Centre, University of Edinburgh, JCMB, The King's Buildings,
Mayfield Road, Edinburgh, EH9 3JZ, UK*

Abstract

We introduce HPCx - the UK's new National HPC Service - which aims to deliver a world-class service for capability computing to the UK scientific community. HPCx is targeting an environment that will both result in world-leading science and address the challenges involved in scaling existing codes to the capability levels required. Close working relationships with scientific consortia and user groups throughout the research process will be a central feature of the service. A significant number of key user applications have already been ported to the system. We present initial benchmark results from this process and discuss the optimisation of the codes and the performance levels achieved. We find a range of performance with some algorithms scaling far better than others.

1. INTRODUCTION

HPCx is the name of the UK's new National High Performance Computing Service. It is a large IBM p690 cluster whose configuration is specifically designed for high-availability capability computing¹. The Engineering and Physical Sciences Research Council (EPSRC) is overseeing the project, on behalf of the UK scientific community. HPCx is a joint venture between the Daresbury Laboratory of the Central Laboratory for the Research Councils (CLRC) and Edinburgh Parallel Computing Centre (EPCC) at the University of Edinburgh. IBM (UK) Ltd has been chosen as the hardware supplier for the six-year duration of the project.

2. THE HPCx PROJECT

The British Government, through EPSRC, has funded the project to approximately £53M (~USD 85M). The scope of the project is to operate and support the principal academic and research high performance computing service for the UK. The principal objective is to provide a capability computing service to run key scientific applications that can only be run on the very highest performing computing platforms.

The project is a collaboration between three partners: CLRC Daresbury Laboratory, Edinburgh Parallel Computing Centre (EPCC) and IBM. HPCx, formally known as UoE HPCx Limited, is a wholly-owned subsidiary of the University of Edinburgh. It is under contract to EPSRC to provide HPC services and it sub-contracts appropriate packages to the consortium partners. The partnership combines two of Europe's leading academic HPC, e-Science and technology transfer centres, with significant experience in operating and supporting national grand challenge and capability services on novel architectures.

CLRC Daresbury Laboratory has been an HPC service provider to the UK academic community for over 25 years - the first Cray 1 in Europe was delivered there in 1978. CLRC's Computational Science and Engineering Department is the UK's premier research, development and support centre for leading-edge academic science and engineering simulation codes. CLRC also provides distributed computing support for COTS processor and network technologies, evaluating scalability and performance [1]. CLRC also hosts the UK Grid Support Centre.

EPCC, located at the University of Edinburgh, was established in 1991 as the University's interdisciplinary focus for high-performance computing and its commercial exploitation. EPCC has hosted specialised HPC services for the UK's QCD community since 1989. The 5Tflop/s QCDOC system is due to be installed in 2003, in a project with Columbia, IBM and Brookhaven National Laboratory. From 1994 until 2002, EPCC

¹ Jobs which use a significant fraction of the total resource.

operated and supported UK national services on Cray T3D and T3E systems.

Within HPCx, CLRC and EPCC both provide "Added Value" services (primarily scientific support, code optimisation, and code development) and operations and system support services. The physical accommodation is provided at CLRC's Daresbury Laboratory. IBM is the technology partner, providing the hardware, system software and appropriate maintenance and support services. They offer world-leading and competitively priced HPC technology, an aggressive road map over the next 6 years and significant resources for science support.

HPCx is a six-year project and the technology will be provided in three phases, with defined performance levels at year 0, year 2 and year 4. The performance targets for the three phases are defined in terms of Linpack Rmax performance. The targets for the three phases are approximately 3 Tflop/s, 6 Tflop/s and 12 Tflop/s, roughly following Moore's Law. Online and offline storage will increase proportionately, with 50, 100 and 200 TB in the three phases. There is a flexible approach to the technology refreshes in phases 2 and 3, and future upgrades are likely to feature the "Federation" switch and POWER5-based architectures.

The focus is on maximising the delivery of capability computing; the service is not intended to be used as a task farm or for multiple modest-sized throughput jobs. There is a comprehensive support service providing porting, optimisation, training and new applications outreach. The service has a 7 x 24 operating regime with high RAS requirements.

3. THE HPCx PHASE 1 SYSTEM

Delivery of the Phase 1 system commenced on 4th October 2002. The full user service opened officially on 9th December 2002, although many users had benefited from up to a month's early access prior to this date. The system comprises 40 Regatta-H SMP compute nodes connected by the "Colony" SP Switch2.

Each shared memory node has 32 1.3GHz POWER4 processors and 32 GB memory giving the system an aggregate memory of 1.28 TB. The POWER4 processor has dual floating point units each of which, through the fused multiply-add instruction, is capable of delivering two results per clock cycle. This gives each processor a peak performance of 5.2 Gflop/s and the whole system a peak of some 6.6 Tflop/s.

In order to increase connectivity to the switch and improve communications throughput, each compute node is configured as four 8-way logical partitions (LPARs). The "Colony" SP Switch2 allows each LPAR to have two connections (PCI adapters) into the switch fabric (dual plane), providing approximately 20 usec latency and 350 MBytes/sec bandwidth. Two additional 16 processor nodes are provided as I/O systems. The system runs AIX version 5, with GPFS for file system

support (18TB EXP500) and HSM for backup and archive to tape storage (35TB LTO tape library).

4. TOWARDS CAPABILITY COMPUTING

Perhaps the key challenge for the HPCx service is to deliver on the capability aspirations of the UK community across a broad spectrum of disciplines. We outline in sections 5-10 below initial progress towards this goal by illustrating the current levels of delivered performance from a number of codes across a variety of disciplines, including materials science, molecular simulation, atomic and molecular physics, molecular electronic structure, computational engineering and environmental science.

Considering a total of 10 application codes, we compare performance on the IBM SP/Regatta-H with that found on a number of platforms from other vendors. These include the Cray T3E/1200E ("Turing") and the SGI Origin 3800/R12k-400 system ("Green"), both operated by CSAR at the University of Manchester, UK, and the SGI O3800/R14k-500 system ("Teras") at SARA, Amsterdam. Also included is the Compaq AlphaServer ES45/1000, the TCS-1 system at Pittsburgh Supercomputing Centre, PSC.

5. MATERIALS SCIENCE

5.1. CRYSTAL, AIMPRO and CASTEP

CRYSTAL [2] permits the calculation of wavefunctions and properties of crystalline systems, using a periodic Hartree-Fock or density functional Kohn-Sham Hamiltonian and various hybrid approximations. The wavefunctions are expanded in atom centred Gaussian type orbitals (GTOs) providing a highly efficient and numerically precise solution with no shape approximation to the density or potential. The code is developed within a long standing collaboration between the Theoretical Chemistry Group at the University of Torino, Italy, and the Computational Materials Science Group at Daresbury Laboratory, and is widely distributed internationally. Details of the code and its applications can be found on the CRYSTAL web pages [2,3].

Recent enhancements to the parallel distributed data version of the code, MPP CRYSTAL, include the incorporation of a somewhat faster, and more numerically stable version of the parallel Jacobi diagonalizer [4]. Further, since it can diagonalize not only real symmetric but also Hermitian matrices, the ScaLAPACK library routines are no longer required. This is advantageous because the latter routines both scale poorly and also, dependent upon the eigenvalue spectrum, may require unacceptably large amounts of replicated memory.

The rationalization of the memory management within the code has continued. All large arrays are now dynamically allocated, and further general purpose Fortran 90 modules are available for more complex data

structures, such as linked lists. On MPP systems disk access is often an expensive process, and so as far as is possible this is now avoided. Data is either recalculated or, for distributed objects, stored in memory, the latter being possible because of the memory capacity typically found on these machines.

Table 1. Time in Wall Clock Seconds for CRYSTAL calculations of Crystalline Crambin on the IBM SP/Regatta-H and SGI Origin 3800/R12k-400 for Three Different Basis Sets.

CPU's	SGI Origin 3800 / R12k-400	IBM SP / Regatta-H
STO-3G Basis (3,948 GTOs)		
32	6559	3238
64	3400	1762
96	2327	1183
128	1810	921
192	1289	682
256	1038	531
384	772	416
512		343
768		278
1024		245
6-31G Basis (7,194 GTOs)		
32	23457	11970
64	12130	6333
96	8240	4248
128	6251	3305
192	4437	2317
256	3496	1801
384	2449	1321
512		1043
768		817
1024		668
6-31G* Basis (12,354 GTOs)		
256		1924
512		1099
1024		716

Timings on the IBM SP/Regatta-H and Origin 3800/R12k-400 of CRYSTAL 2000 for a benchmark

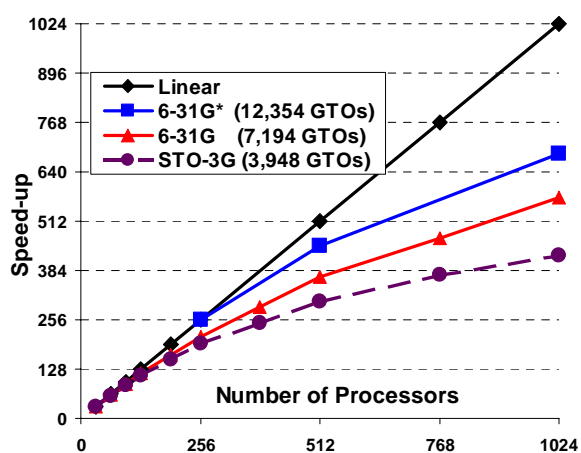


Figure 1. Scalability of CRYSTAL-2000 in Calculations on Crystalline Crambin.

calculation on crystalline crambin [5] with 1284 atoms per cell are reported in Table 1. These calculations were performed in basis sets of increasing quality, with reported times for the STO-3G (3,948 GTOs) and 6-31G (7,194 GTOs) basis sets referring to 3 cycles of the iterative SCF process, and the times for the 6-31G** basis (12,354 GTOs) being for a single SCF iteration. The corresponding speedups on the IBM SP/Regatta-H shown in Figure 1 reveal excellent scalability that is enhanced with improvements in the basis set.

AIMPRO (Ab Initio Modelling PROgram) was written by Patrick Briddon et al. at Newcastle University [6]. It can be used to study both molecules and 3D periodic systems using a Gaussian basis set.

The AIMPRO benchmark models a carbon impurity in a silicon lattice; 216 atoms are present in total (1 carbon, 215 silicon) and four k points are used, with a total of 5180 basis functions. The major operation to be performed in the calculation is a diagonalization of size the number of basis functions in the benchmark.

Performance figures on the IBM/SP Regatta-H (HPCx) and SGI Origin 3800/R12k-400 are shown in Figure 2. These suggest that the IBM / SGI Origin performance ratio of 4.3 up to 64 processors falls beyond this CPU count, to 2.3 on 128 and only 1.6 on 256 processors.

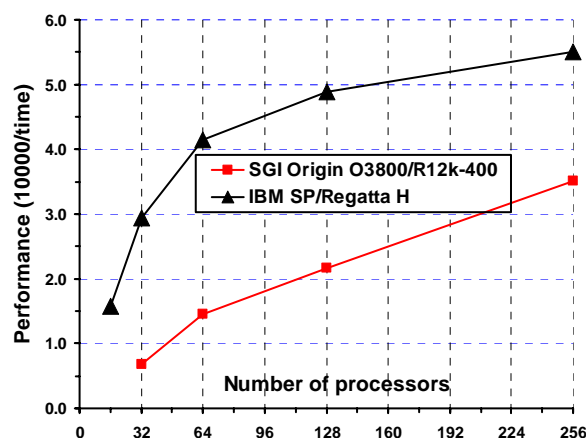


Figure 2. The AIMPRO Benchmark

The major performance-related problem with AIMPRO is the diagonalizations it has to perform. These are conducted at present using the ScaLAPACK routine PDSYEVX. This routine uses the QR method which, while well suited for AIMPRO in that only a subset of the eigenvalues need be calculated, does not scale well to high processor count.

CASTEP [7] is an *ab initio* computational materials code using the plane-wave pseudopotential method. It has been developed by the UKCP Consortium and is distributed commercially by Accelrys Plc. In common with many plane-wave based codes, CASTEP uses 3-dimensional FFTs to transform between real-space and wave-space. In order to perform the FFTs efficiently on each processor, there is a data transformation step using the MPI collective, MPI_Alltoallv. Given the latency dependency of this collective, it is clear that this step

will be critically dependent on interconnect, and will not scale as well on, for example, the IBM SP as on the Cray T3E/1200E. Where a case has more than 1 k-point it is important to use “kG” parallelization to minimize the number of processors involved in calls to MPI_Alltoallv; the fewer the number of k-points, the greater the number of processors involved (finally leading to “G” parallelization).

Table 2. Time in Wall Clock Seconds for the 8k and 32-kpoint CASTEP TiN Benchmark Calculations.

CPU's	Cray T3E/ 1200E	SGI O3800 R14k-500	Compaq AlphaServer ES45/1000	IBM SP / Regatta-H Cheetah
TiN, 1x1x1, 8 k-points				
32	6038	2287	1312	1149
64	3109	1096	800	610
128	1712	536	560	577 [†]
TiN, 1x1x1, 32 k-points				
32	22561			4816
64	13033			2621
128	7244			1403

[†] 404secs currently on the HPCx system

The first CASTEP benchmark reported is an 8 k-point total energy calculation of a 33-atom slab of TiN, using the Vanderbilt ultrasoft pseudo-potential. Employing “kG” parallelisation, the benchmark involves a total of 88,000 plane waves (ca. 11,000 per k-point) and a 3D FFT grid size of 108x36x36. Convergence was achieved in 30 SCF cycles using the Pulay density mixing minimiser scheme. In this case, the “kG” parallelization divides the processors into 8 groups, each dealing with one k-point. The 3D FFTs are then distributed over processors within a group rather than across all processors so that ideally 8 independent 3D FFTs may be performed simultaneously. Timings are reported (Table 2) on the Cray T3E/1200E, Compaq AlphaServer SC ES45/1000, IBM SP/Regatta-H and SGI Origin 3800/R14k-500. A second benchmark involved applying a dense Monkhorst Pack (MP) mesh to the 8 k-point test case above, leading to 32 k-points. We label these two benchmarks as “1x1x1-8” and “1x1x1-32” respectively.

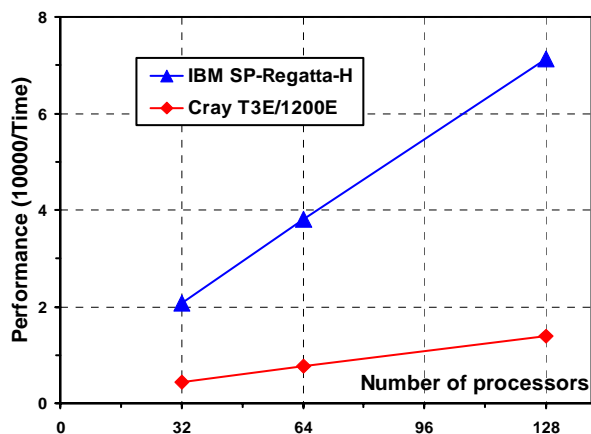


Figure 3. CASTEP TiN Benchmark, 32 k-points (1x1x1-32)

The performance of CASTEP varies greatly with the size of the problem and the number of k-points. With few k-points and large numbers of processors MPI_Alltoallv dominates, so that the overall performance on the IBM SP scales far less effectively than on the Cray T3E/1200E. For the 1x1x1-8 case, respectable IBM / Cray T3E performance ratios of around 5.6 (32 processors) and 5.0 (64 processors) appear to fall rapidly beyond this CPU count, to only 3.0 on 128 processors (1x1x1-8, Table 2). In contrast, the performance of the 32 k-point case remains respectable with ratios of around 4.7, 4.9 and 5.1 on 32, 64 and 128 processors respectively (Figure 3, 1x1x1-32). The 128 processor run takes around 1,403 seconds on the IBM SP/Regatta-H. Large numbers of k-points are typical for calculations of metals.

Two potential developments are being undertaken to look at improving the current CASTEP performance. It is possible to call MPI_Alltoallv for several FFTs at once, thus reducing the number of message passing calls. Initial investigation suggests that while this does improve matters, it is only a major improvement for G parallelism, at least for the test cases. Secondly, use may be made of the additional memory available on the IBM SP to avoid some of the FFTs. In the program a number of these are repeated, the reason being memory constraints on older machines, and so on more modern machines this may be avoided. Initial investigations show a similar improvement as above for kG and G parallelization. Recent improvements in the communications have led to the 128-processor 1x1x1-8 time reducing from 577 to 404 secs.

What is clear from the above benchmarks is the limited scalability likely to arise on the HPCx system in any application that involves global communication routines (CASTEP), or a dependency on linear algebra routines with extensive communication requirements (AIMPRO). This comes as little surprise given the known limitations of the present Colony-based interconnect.

6. MOLECULAR SIMULATION

6.1. DL_POLY and NAMD

DL_POLY [8] is a general-purpose molecular dynamics simulation package designed to cater for a wide range of possible scientific applications and computer platforms, especially parallel hardware. Two graphical user interfaces are available, one based on the CERIU² Visualiser from Accelrys and the other on the Java programming language.

DL_POLY supports a wide range of application areas, including [9] ionic solids, solutions, metals, zeolites, surfaces and interfaces, complex systems (e.g. liquid crystals), minerals, bio-systems, and those in spectroscopy. Comprehensive benchmarking of the replicated data (RD) version (Version 2.11) of DL_POLY [10,11] clearly reveals the limitations inherent in the RD strategy, with restrictions in the size

of system amenable to study, and limited scalability on current high-end platforms. These limitations apply not only to systems possessing complex molecular topologies and constraint bonds, but also to systems requiring simple atomic descriptions, systems that historically exhibited excellent scaling on the Cray T3E/1200E. Significant enhancements to the codes capabilities have arisen from the recent release of the distributed data (or domain decomposition) version (DL_POLY 3), developments that have been accelerated in light of the arrival of the HPCx system.

Evaluation of the Coulomb potential and forces in DL_POLY is performed using the smooth particle mesh Ewald (SPME) algorithm [12]. As in all Ewald [13] methods, this splits the calculation into two parts, one performed in real space and one in Fourier space. The former only requires evaluation of short ranged functions, which fits in well with the domain decomposition used by DL_POLY 3, and so scales well with increasing processor count. However the Fourier component requires 3 dimensional FFTs to be performed. These are global operations and so a different strategy is required if good scaling is to be achieved.

The original implementation involved replicating the whole FFT grid on all processors and performing the FFTs in serial after which each processor could evaluate the appropriate terms for the atoms that it held. This method clearly has a number of well known drawbacks.

While both open source 3D parallel FFTs (such as FFTW [14]) and proprietary routines (such as Cray's PCCFFT) are available, neither adequately address all the issues. The problem is that they impose a data distribution, typically planes of points, that is incompatible with DL_POLY's spatial domain decomposition, so while a complete replication of the data is not required, it is still necessary to perform extensive data redistribution which will limit the scaling of the method.

To address these limitations, a parallel 3D FFT has been written [15] which maps directly onto DL_POLY's data distribution; this involved parallelizing the individual 1D FFTs in an efficient manner. While the method will be slower than the proprietary routines for small processor counts, at large numbers it is attractive, since (a) while moving more data in total, the method requires much fewer messages, so that in the latency dominated regime it should perform better, and (b) global operations, such as the *all to all* operations used in both FFTW and PCCFFT, are totally avoided. More generally the method is extremely flexible, allowing a much more general data distribution than those of other FFTs, and as such should be useful in other codes which do not map directly onto a "by planes" distribution.

In the present section we present recent results obtained on the IBM SP/Regatta-H, Compaq AlphaServer ES45/1000 and SGI Origin 3800/R14k-500, results which highlight the drastic improvements in both system size and performance made possible through these developments. The four benchmarks reported in Table 3 include two Coulombic-based simulations of

NaCl, one with 27,000 ions, the second with 216,000 ions. Both simulations involve use of the Particle Mesh Ewald Scheme, with the associated FFT treated by the algorithm outlined above [15] in which the traditional all-to-all communications are replaced by the scheme that relies on column-wise communications only. The reported timings are for 500 time steps in the smaller calculation, and 200 time steps in the larger simulation.

The other two benchmarks are macromolecular simulations based on Gramicidin-A; the first includes a total of 99,120 atoms and 100 time steps. The second, much larger simulation, is for a system of eight Gramicidin-A species (792,960 atoms), with the timings reported for just 50 time steps. In terms of time to solution, we see that the AlphaServer SC outperforms the Origin 3800 at all processor counts in all four benchmarks; both 256 CPU runs for the larger NaCl and Gramicidin-A simulations suggest factors of 1.3-1.4.

These results show a marked improvement in performance compared to the replicated data version of the code, with the gratifying characteristic of enhanced scalability with increasing size of simulation, both in the ionic and macromolecular simulations. Considering the NaCl simulations, we find speedups of 139 and 122 respectively on 256 processors of the Origin 3800 and AlphaServer SC in the 27,000-ion simulation. These figures increase to 172 and 171 respectively in the larger simulation featuring 216,000 ions. While the total times to solution on the IBM SP are broadly in line with the

Table 3. Time in Wall Clock Seconds for four DL_POLY 3 benchmark Calculations on the Compaq AlphaServer SC ES45/1000, IBM SP/Regatta-H and SGI Origin 3800.

CPUs	SGI Origin 3800 / R14k-500	Compaq Alpha ES45 / 1000	IBM SP / Regatta-H
NaCl ; 27,000 ions, 500 time steps			
16	313	183	160
32	168	103	97
64	92	57	61
128	53	37	42
256	36	24	
NaCl; 216,000 ions, 200 time steps			
16	764	576	455
32	387	326	234
64	201	168	128
128	116	91	78
256	71	54	48
512			41
Gramicidin A; 99,120 atoms, 100 time steps			
16	282	173	166
32	167	109	107
64	100	75	74
Gramicidin A; 792,960 atoms, 50 time steps			
32	661	370	349
64	273	186	189
128	140	109	114
256	97	68	77
512			56

AlphaServer SC, the scalability remains inferior.

For these systems the FFT routine remains the poorest scaling; the time required to exchange scales poorly on the IBM SP with increased processor count. Doubling the number of processors roughly halves the amount of data a processor has to send, so one would expect the time to roughly halve. In practice this is not the case, and at certain processor counts the time dramatically increases. This can be traced to the message passing occurring via the switch rather than through shared memory. For example, for the x direction, which never has to use the switch in the present case, the scaling is reasonable, while for the z direction, which has to use the switch for the first time at 16 processors, there is a major spike in the timings.

A more compelling improvement with system size is found in the macromolecular Gramicidin-A simulations. Again the SPME algorithm is used for evaluation of the Coulomb field, but in these simulations there is the extra complication of constraints on the atoms' motions, which reflects chemical bonds in the system. The shake algorithm is used to evaluate the constraints, and this is again potentially a global operation and so, as for the FFT, good scaling is difficult to achieve. In the distributed data implementation, both SHAKE and short-range forces require only nearest neighbour communications, suggesting that communications should scale linearly with the number of nodes, in marked contrast to the replicated data implementation. This is borne out in practice. In the larger simulation (with 792,960 atoms) we find speedups of 218 and 175 on 256 processors of the Origin 3800 and AlphaServer SC respectively. This level of scalability represents a significant advance over that exhibited by both DL_POLY 2 and CHARMM [10].

Table 4: Elapsed Time per time step (seconds) for the NAMD ApoA-I Benchmark on the IBM SP/Regatta H and Compaq AlphaServer SC ES45/1000.

Number of processors	Compaq Alphaserver SC ES45/1000	IBM SP/Regatta-H
1	7.86	7.97
2		4.47
4		2.19
8		1.11
16		0.580
32		0.305
64		0.163
128	0.0715	0.0985
256	0.0403	0.0664
512	0.0239	0.0424
1024	0.0176	0.0252

NAMD is the parallel, object-oriented molecular dynamics program designed for high performance simulations of large biomolecular systems [16]. NAMD employs the prioritized message-driven execution capabilities of the Charm++/Converse parallel runtime

system, allowing excellent parallel scaling on both massively parallel supercomputers and commodity-based workstation clusters. An initial implementation of the code on HPCx was carried out by Rik Kufryn (NCSA), and Table 4 shows the elapsed time per time step for the standard NAMD ApoA-I benchmark [17], a system comprising 92,442 atoms, with 12Å cutoff and PME every 4 time steps, periodic. Also shown are the timings from the optimised version of the code running on PSC's AlphaServer SC ES45/1000. These show that the AlphaServer marginally outperforms the IBM SP on a single CPU, and with a speed-up of 447 on 1024 processors, exhibits superior scalability compared to the IBM SP (a corresponding speed-up of 316). This is to be expected in light of the superior interconnect associated with the AlphaServer. We would expect these scalability figures to improve with larger simulations in light of performance attributes demonstrated on the TCS-1 system at Pittsburgh (e.g. a speedup of 778 on 1024 CPUs in a 327K particle simulation of F₁-ATPase).

7. ATOMIC & MOLECULAR PHYSICS

7.1. H2MOL and PFARM

H2MOL [18] is a code from the Multiphoton and Electron Collision Consortium (CCP2) written by Ken Taylor & Daniel Dundas, Queens University Belfast. It solves the time-dependent Schrödinger equation to produce estimates of energy distributions for laser-driven dissociative ionization of the H₂ molecule. A cylindrical computational grid is defined with ϕ , ρ and Z coordinates, with the Z domain distributed amongst an array of processors arranged logically in a triangular grid (to take advantage of symmetry). A feature of the way this code has been written is that it specifies as constant the number of grid points **per processor** in the z -direction. Thus with increasing numbers of processors it is working with an increasingly refined mesh i.e., for this benchmark, perfect scaling would be represented by a flat timing profile across the different processor counts.

Table 5: Total Elapsed Times (seconds) for the fixed-point H2MOL Benchmark (ϕ points = 11, ρ points = 30 and Z points = 11 (per processor)).

Number of processors	Cray T3E/1200E	IBM SP / Regatta-H	(Tasks,T) X (Nodes,N)
6	5650	2098	6T x 1N
15	5935	2343	5T x 3N
28	6155		
45	6396		
66	6622	2660	6T x 11N
91	7123		
120	7198	3353	8T x 15N*
231	7742	2550	6T x 20N
325		3004	7T x 33N
435		3071	5T x 65N
496	†	3269	5T x 87N
	†	3952*	8T x 62N*

† Memory Exceeded

The results of Table 5 refer to a problem definition with ϕ points = 11, ρ points = 30 and Z points = 11 (per processor). This is the maximum problem size that the restricted memory on the Cray-T3E/1200E (256 MByte) can accommodate.

The IBM SP / Cray T3E performance ratio starts at around 3 on 6 processors and reduces to just above 2.5 on 231 processors. For calculations involving low numbers of processors, performance is ESSL / LIBSCI intensive (highly optimized ZGEMM, ZAXPY, ZDOT routines) and corresponds to around a third of peak for both IBM and Cray machines. Therefore the best relative performance to be expected on low processor counts reflects the peak ratios of the two machines i.e. 5.2 Gflops (IBM SP) / 1.2 Gflops (Cray T3E) = 4.3. By 120 processors (8 tasks on 15 nodes) there is no more than a factor of two performance gain from the IBM SP. At 231 processors, with partially occupied nodes (7 tasks on each of 33 nodes), the performance ratio has improved to around 2.5. Table 5 demonstrates that the timings on large processor counts of HPCx are fairly flat, representing reasonable scaling. Runs undertaken with fully occupied L-PAR'd nodes (*) are slower than those with at least one free task per node.

Fixing a global number of Z points, rather than a local number of Z points, allows the scaling of the code to be examined more clearly. The results of Figure 4 involve the ϕ and ρ parameters being set as above, with the total (global) number of Z points remaining fixed for different sized processor arrays. Hence, each run is now computing an identical problem. In the graph below, the number of timesteps has been reduced to increase throughput (this has no impact on scaling).

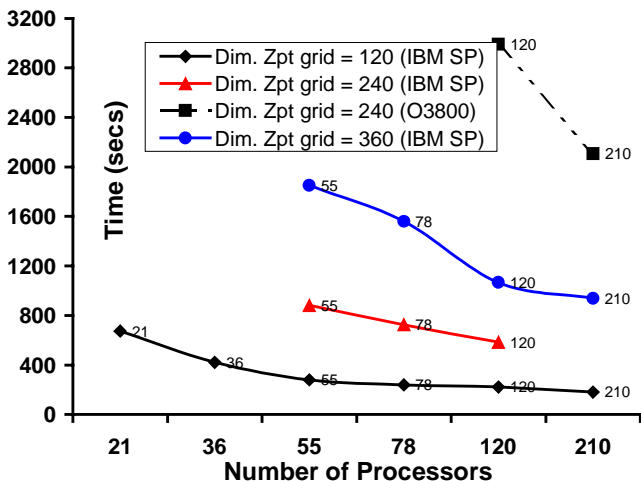


Figure 4. The H2MOL Benchmark with Fixed Problem Size

Evidently, the scalability improves as the problem size increases. Currently, users of the code are investigating problems with over 400 Z points on the SGI Origin 3800 at CSAR.

PFARM: In recent years new high performance codes and techniques have been developed to extend the successful R-matrix formalism to treat applications such as the description of the edge region in Tokamak

plasmas (fusion power research) and for the interpretation of astrophysical spectra. The parallel R-matrix program PFARM has been optimised to treat open d-shell atoms and ions as well as intermediate energy scattering problems [19]. Several new facilities have been incorporated into PFARM to improve its efficiency for these calculations. In electron-ion scattering calculations it is necessary to perform separate computations at a very large number of scattering energies in order to compute the thermally averaged

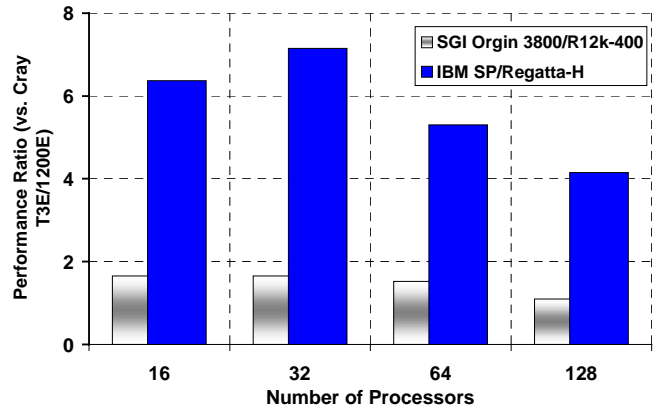


Figure 5. The PFARM Benchmark

collision strengths required in applications. The program splits this energy mesh into a fine region (below all target thresholds) and coarse region (above all target thresholds). By treating each region separately, optimisations specific to each can be made. The parallel calculation has been designed to take advantage of the near-optimal performance of serial BLAS 3 routines; this operation is the computational rate-determining step. Therefore the effect of factors such as matrix dimension, matrix transposes and symmetry have been studied closely, and data is now arranged to maximise the BLAS 3 performance [19]. The major outstanding efficiency issues now relate to the required matrix diagonalisations (using the PeIGS parallel eigensolver [20]) and to load balancing. To address the latter problem, automated load balancing that adapts to each specific application has been developed. A computational model of the parallel calculation has been constructed that describes the relative speed of each component of the functional decomposition. Factors such as physical properties, hardware, relative BLAS/LAPACK performance and channel-splitting are all included in the model. The model is incorporated into a controlling Perl script that can predict dynamically an optimum configuration of processors for a particular physical problem. For each computation, processor configurations and the associated timing data is saved in order to refine the model for future runs.

The code has been ported to the IBM SP/Regatta-H and SGI Origin. Figure 5 shows the performance of the IBM SP and Origin 3800/R12k-400 relative to the Cray T3E/1200E for a typical Ni^{3+} scattering problem. This figure suggests that the IBM SP outperforms the SGI Origin 3800 and Cray T3E/1200E by factors of 3.8 and 4.2 on 128 processors.

8. MOLECULAR ELECTRONIC STRUCTURE

8.1. GAMESS-UK

GAMESS-UK [21] represents a typical established electronic structure code, comprising some 800K lines of Fortran that permits a wide range of computational methodology to be applied to molecular systems. Improving the scalability of the parallel code has been addressed in part by adopting a number of the tools developed by the High Performance Computational Chemistry (HPCC) group from the Environmental Molecular Sciences Laboratory at PNNL, in Richland, Washington. These include the Global Array (GA) toolkit [22] that provides an efficient and portable "shared-memory" programming interface for distributed-memory computers, and the scalable, fully parallel eigensolver, PeIGS whose numerical properties satisfy the needs of the chemistry applications [20].

Table 6. Time in Wall Clock Seconds for Four GAMESS-UK Benchmark Calculations on the Compaq AlphaServer SC ES45/1000, IBM SP/Regatta-H and SGI Origin 3800/R14k-500.

CPU's	SGI Origin 3800 / R14k- 500	Compaq Alpha ES45 / 1000	IBM SP / Regatta-H
Cyclosporin (1000 GTOs) DFT/B3LYP 6-31G			
32	1191	713	666
64	704	424	424
128	481	310	322
Cyclosporin (1855 GTOs) DFT/B3LYP 6-31G**			
32	4731	2504	2614
64	2838	1584	1681
128	1867	1100	1281
Valinomycin (882 GTOs) DFT/HCTH			
32	2306	1301	1329
64	1228	705	749
128	734	415	493
(C ₆ H ₄ (CF ₃) ₂) ₂ SCF 2nd Derivatives			
16	1490		860
32	803	501	621
64	494	360	371
128		246	213

The main source of parallelism in the DFT module is the computation of the one- and two-electron integrals together with the exchange correlation contributions, and their summation into the Fock matrix. The computation of these quantities is allocated dynamically using a shared global counter. With the capabilities afforded by the GA tools [22], some distribution of the linear algebra becomes trivial. As an example, the SCF convergence acceleration algorithm (DIIS - direct inversion in the iterative subspace) is distributed using GA storage for all matrices, and parallel matrix multiply and dot-product functions. This not only reduces the time to perform the step, but the use of distributed memory storage (instead of disk) reduces the need for I/O during the SCF process.

Diagonalisation of the resulting Fock matrix is now based on the PeIGS module from NWChem [23].

Substantial modifications were required to enable the SCF 2nd derivatives [24] to be computed in parallel. The conventional integral transformation step has been omitted, with the SCF step performed in direct fashion and the MO integrals, generated by re-computation of the AO integrals, and stored in the global memory of the parallel machine. The GA tools manage this storage and subsequent access. The basic principle by which the subsequent steps are parallelised involves each node computing a contribution to the current term from MO integrals resident on that node. For some steps, however, more substantial changes to the algorithms are required. The coupled Hartree-Fock (CPHF) step and construction of perturbed Fock matrices are again parallelised according to the distribution of the MO integrals. The most costly step in the serial 2nd derivative algorithm is the computation of the 2nd derivative two-electron integrals. This step is trivially parallelised through a similar approach to that adopted in the direct SCF scheme - using dynamic load balancing based on a shared global counter. In contrast to the serial code, the construction of the perturbed Fock matrices dominates the parallel computation. It seems almost certain that these matrices would be more efficiently computed in the AO basis, rather than from the MO integrals as in the current implementation, thus enabling more effective use of sparsity when dealing with systems comprising more than 25 atoms.

The performance of the DFT and SCF 2nd Derivative modules on the SGI O3800/R14k-500, Compaq AlphaServer SC ES45/1000 and IBM SP/Regatta-H are shown in Table 6. Note that the DFT calculations did not exploit CD fitting, but evaluated the coulomb matrix explicitly. Considering the DFT results, modest speedups of 81, 73 and 65 are obtained on 128 processors of the Origin 3800, AlphaServer SC and IBM SP respectively for the larger cyclosporin calculation. Somewhat better scalability is found in the Valinomycin DFT calculation where a greater proportion of time is spent in integral evaluation arising from the more extended basis sets [25]; speedups of 101, 100 and 86 respectively are obtained on 128 processors of the Origin, AlphaServer and IBM SP. The enhanced performance of the SCF 2nd Derivative module on the IBM SP/Regatta H arises from the decreased dependency on latency exhibited by the current implementation compared to the DFT module. Thus the timings of Table 6 suggest that the SP is outperforming the AlphaServer SC on 128 processors.

The less than impressive scalability of both GAMESS-UK (and also NWChem) on the IBM SP arises to some extent from the dependency of both codes on a Global Array implementation that is based on IBM's LAPI communication library [26]. The current implementation of LAPI on POWER4-based architectures is far from optimal, with the measured latencies and bandwidths significantly inferior to those measured on corresponding POWER3-based systems. We are currently working with PNNL and IBM's LAPI

team to further understand and address these shortcomings.

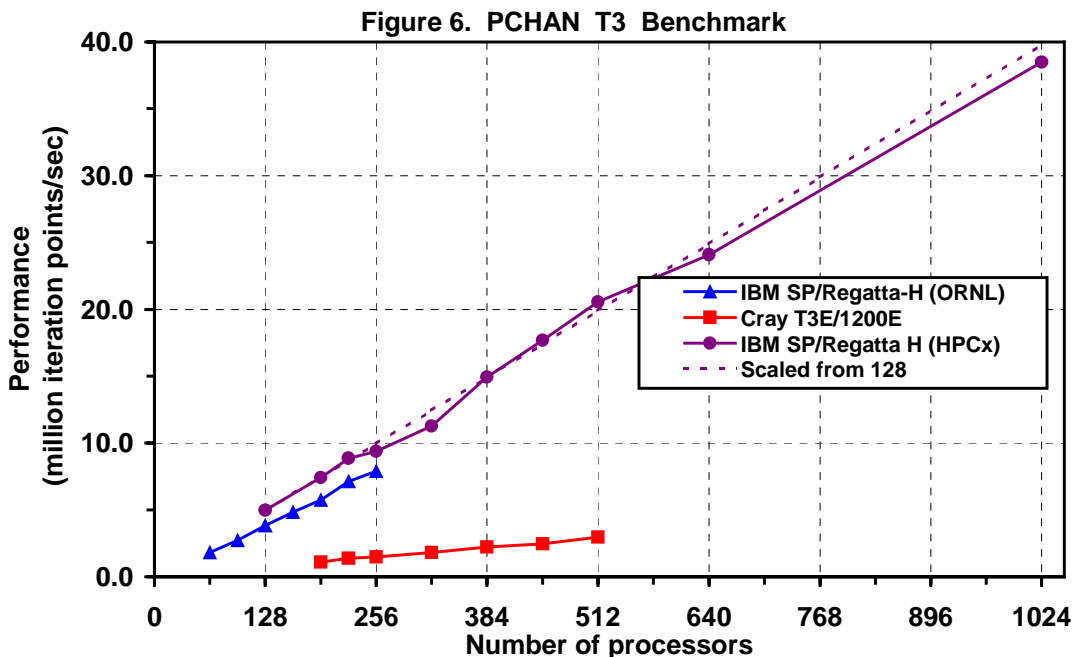
9. Computational Engineering

9.1. PCHAN

Fluid flows encountered in real applications are invariably turbulent. There is, therefore, an ever-increasing need to understand turbulence and, more importantly, to be able to model turbulent flows with improved predictive capabilities. As computing technology continues to improve, it is becoming more feasible to solve the governing equations of motion – the Navier-Stokes equations – from first principles. The direct solution of the equations of motion for a fluid, however, remain a formidable task and simulations are only possible for flows with small to modest Reynolds numbers. Within the UK, the Turbulence Consortium (UKTC) has been at the forefront of simulating turbulent flows by direct numerical simulation (DNS). UKTC has

on the IBM SP/Regatta-H systems, for which we have data from both the Cheetah system (at ORNL) and from the HPCx system. Where benchmark runs allow direct processor-for-processor comparison between the IBM and Cray systems (192-512 processors), the performance ratio is fairly constant at around 6.2 to 7.2. e.g., the 512 CPU T3 Benchmark required 227 elapsed seconds, with the HPCx system outperforming the Cray T3E/1200E (1,575 seconds) by a factor of 6.94. Figure 6 shows performance results from the T3E and the two IBM systems. The HPCx runs incorporate code optimisations, which account for a 20%-30% improvement; these were not present in the Cheetah implementation.

The most important communications structure within PCHAN is a halo-exchange between adjacent computational sub-domains. Providing the problem size is large enough to give a small surface area to volume ratio for each sub-domain, the communications costs are small relative to computation and do not constitute a bottleneck. Code optimisation for the POWER4 cache architecture has shown to be highly beneficial in



developed a parallel version of a code to solve problems associated with shock/boundary-layer interaction. The code (SBLI) was originally developed for the Cray T3E and is a sophisticated DNS code that incorporates a number of advanced features: namely high-order central differencing; a shock-preserving advection scheme from the total variation diminishing (TVD) family; entropy splitting of the Euler terms and the stable boundary scheme [27]. The code has been written using standard Fortran 90 code together with MPI in order to be efficient, scalable and portable across a wide range of high-performance platforms.

The PCHAN benchmark is a simple turbulent channel flow benchmark using the SBLI code. Performance with the T3 Grid Benchmark data case (360x360x360) shows close to ideal scaling on both the Cray T3E/1200E and

increasing the performance.

10. Environmental Science

10.1. POLCOMS

The Proudman Oceanographic Laboratory Coastal Ocean Modeling System (POLCOMS) has been developed to tackle multi-disciplinary studies in coastal/shelf environments [28]. The central core is a sophisticated 3-dimensional hydrodynamic model that provides realistic physical forcing to interact with, and transport, environmental parameters.

The hydrodynamic model is a 4-dimensional finite difference model based on a latitude-longitude Arakawa B-grid in the horizontal and S-coordinates in the vertical.

Conservative monotonic PPM advection routines are used to ensure strong frontal gradients. Vertical mixing is through turbulence closure (Mellor-Yamada level 2.5).

In order to study the coastal marine ecosystem, the

3800/R12k-400 systems operated by CSAR at the University of Manchester, UK.

We find, as expected, that, as the grid size increases, the ratio of communication to computation in the code

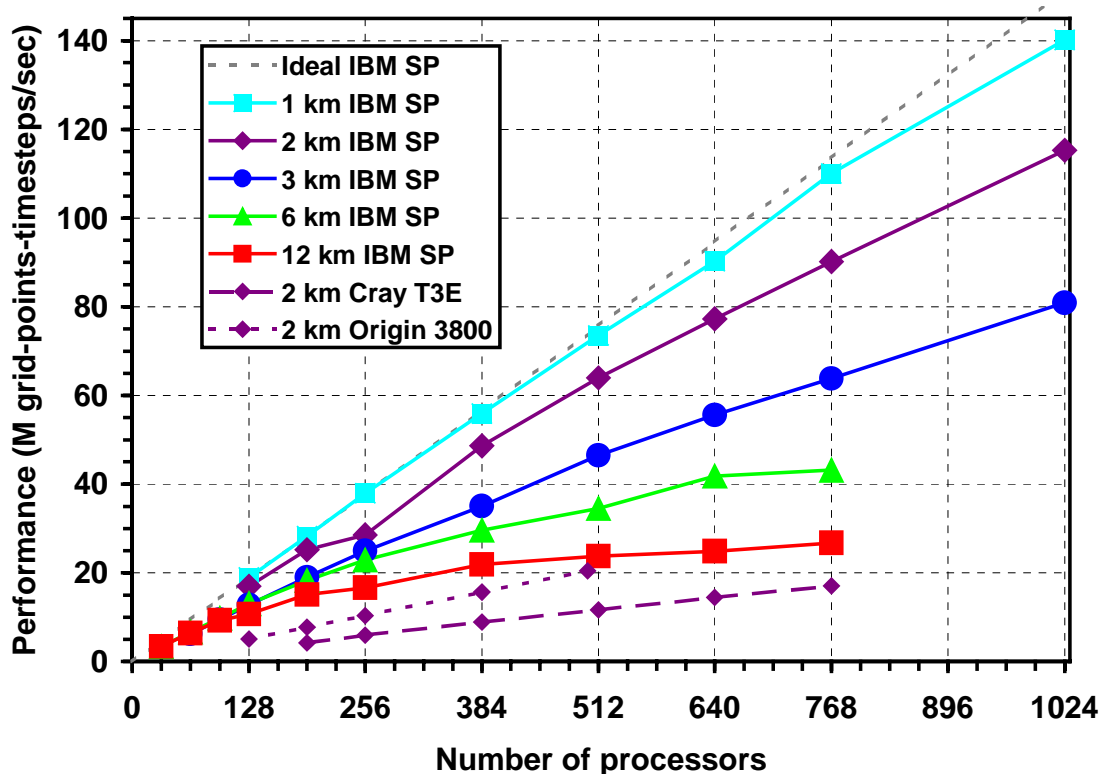


Figure 7. Performance of the POLCOMS Hydrodynamics code as a Function of Grid Size.

POLCOMS model has been coupled with the European Seas Regional Ecosystem Model (ERSEM) [29]. Studies have been carried out, with and without the ecosystem sub-model, using a shelf-wide grid at 12km resolution. This results in a grid size of approx. 200 x 200 x 34. In order to improve simulation of marine processes, we need accurate representation of eddies, fronts and other regions of steep gradients. The next generation of models will need to cover the shelf region at approximately 1km resolution.

In order to assess the suitability of the POLCOMS hydrodynamic code for scaling to these ultra-high resolutions we have designed a set of benchmarks which runs (without the ecosystem model) at grid sizes representative of resolutions from the current 12km down to 1km. The resolutions (and horizontal grid dimension) chosen are 12km (200), 6km (400), 3km (800), 2km (1200) and 1km (2400). The number of vertical levels was fixed at 34. In order to keep benchmark run times manageable, the runs were kept short (100 timesteps) and the initialisation and finishing times were subtracted from the total run time. So as to compare properly runs with different grid sizes, performance is reported in Figure 7 as the amount of work (gridpoints x timesteps) divided by the time. Runs using up to 1024 processors of the HPCx system are shown compared to the Cray T3E-1200E and Origin

improves and so does the scalability. At 2km resolution the code is scaling almost linearly on all three systems with the HPCx system delivering approx. 5.5 times the performance of the Cray T3E and 3.1 times the Origin 3800.

11. Summary

We have introduced HPCx, the UK's new National HPC Service which aims to deliver a world-class service for capability computing to the UK scientific community. HPCx is targeting an environment that will both result in world-leading science and address the challenges involved in scaling existing codes to the capability levels required.

A significant number of key user applications have already been ported to the system. The initial benchmark results from this process and the performance levels achieved have highlighted a wide range of performance, with some algorithms scaling far better than others. What is clear is that the current limitations, arising in the main from inadequacies associated with the Colony switch, demand a major focus on algorithm development designed to remove existing dependencies on collective, global operations. Where this has been addressed e.g., CRYSTAL, PCHAN and POLCOMS, we find excellent levels of scalability and performance.

12. References

- [1] *Computational Chemistry Applications: Performance on High-End and Commodity-class Computers* M. F. Guest and P. Sherwood. Proceedings of HPCS 2002, Moncton, Canada, 2002
- [2] See www.cse.dl.ac.uk/Activity/CRYSTAL, www.chimifm.unito.it/teorica/crystal
- [3] *Ab Initio Modelling in Solid State Chemistry*, European Summerschool, MSSC2002, 8-13 Sept. 2002, Torino, Italy <http://www.chimifm.unito.it/teorica/mssc2002>.
- [4] <http://www.cse.clrc.ac.uk/arc/bfg.shtml>
- [5] The structure of Crambin is derived from XRD data at 0.52 Å. This crystal structure contains 1284 atoms.
- [6] AIMPRO, S. Öberg, J. Goss, R. Jones, and P.R. Briddon.
- [7] *Iterative Minimization Techniques for Ab Initio Total Energy Calculations: Molecular Dynamics and Conjugate Gradients*, M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, J. D. Joannopoulos, *Rev. Mod. Phys.* **64** (1992) 1045.
- [8] *DL_POLY: A general purpose parallel molecular dynamics simulation package*, W Smith and T R Forester, *J. Molec. Graphics* **14** (1996) 136.
- [9] *DL_POLY: Applications to Molecular Simulation*, W Smith, C Yong and M Rodger, *Molecular Simulation* **28** (2002) 385.
- [10] *Application Performance on High-end and Commodity-class Computers*, M.F. Guest, <http://www.ukhec.ac.uk/publications>
- [11] W. Smith, http://www.dl.ac.uk/TCSC/Software/DL_POLY/main.html
- [12] Darden et al., *J. Chem.Phys.* **103** (1995) 19.
- [13] P.P. Ewald, *Ann. Physik.* **64** (1921) 253.
- [14] Frigo and Johnson, ICASSP Conference Proceedings (1998).
- [15] <http://www.cse.clrc.ac.uk/arc/fft.shtml>
- [16] *NAMD2: Greater scalability for parallel molecular dynamics*, L. Kalé, R. Skeel, M. Bhandarkar, R. Brunner, A. Gursoy, N. Krawetz, J. Phillips, A. Shinozaki, K. Varadarajan, and K. Schulten. *J. Comp. Phys.*, **151** (1999) 283-312.
- [17] <http://www.ks.uiuc.edu/Research/namd/performance.html>
- [18] Dundas D, Meharg K, McCann J F and Taylor K T, to be published
- [19] *A parallel R-matrix program PRMAT for electron-atom and electron-ion scattering calculations*, A.G. Sunderland, C.J. Noble, V.M. Burke and P.G. Burke, *Comput. Phys. Commun.*, **145** (2002) 311.
- [20] *Parallel inverse iteration with reorthogonalization*, G. Fann and R.J. Littlefield in: *Sixth SIAM Conference on Parallel Processing for Scientific Computing (SIAM)*, (1993) pp.409-413.
- [21] GAMESS-UK is a package of ab initio programs written by M.F. Guest, J.H. van Lenthe, J. Kendrick, K. Schoeffel and P. Sherwood, with contributions from R.D. Amos, R.J. Buenker, M. Dupuis, N.C. Handy, I.H. Hillier, P.J. Knowles, V. Bonacic-Koutecky, W. von Niessen, R.J. Harrison, A.P. Rendell, V.R. Saunders, and A.J. Stone. The package is derived from the original GAMESS code due to M. Dupuis, D. Spangler and J. Wendoloski, NRCC Software Catalog, Vol. 1, Program No. QG01 (GAMESS), 1980.
- [22] *A non-uniform memory access programming model for high-performance computers*, J. Nieplocha, R.J. Harrison, and R.J. Littlefield, 1996, *Global Arrays: J. Supercomputing* **10** (1996) pp.197-220.
- [23] M.F. Guest, E. Apra, D.E. Bernholdt, H.A. Fruechtl, R.J. Harrison, R.A. Kendall, R.A. Kutteh, X. Long, J.B. Nicholas, J.A. Nichols, H.L. Taylor, A.T. Wong, G.I. Fann, R.J. Littlefield and J. Nieplocha, *Future Generation Computer Systems* **12** (1996) pp. 273-289.
- [24] *A Parallel Second-Order Møller Plesset Gradient*, G. D. Fletcher, A. P. Rendell and P. Sherwood. *Molecular Physics.* **91** (1997) 431.
- [25] N. Godbout, D. R. Salahub, J. Andzelm and E. Wimmer, *Can. J. Chem.* **70**, (1992) 560.
- [26] *Performance and Experience with LAPI – a New High-Performance Communication Library for the IBM RS/6000 SP*, G. Shah, J. Nieplocha, J. Mirza, C. Kim, R.J. Harrison, R.K. Govindaraju, K. Gildea, P. DiNicola, C. Bender, *Supercomputing* 2002.
- [27] *Direct Numerical Simulation of Shock/Boundary Layer Interaction*, N. D. Sandham, M. Ashworth and D. R. Emerson, <http://www.cse.clrc.ac.uk/ceg/sbli.shtml>
- [28] *Coupled Marine Ecosystem Modelling on High-Performance Computers*, M. Ashworth, R. Proctor, J.T. Holt, J.I. Allen, and J.C. Blackford in *Developments in Teracomputing*, eds. W. Zwiefelhofer and N. Kreitz, 2001, pp. 150-163, (World Scientific).
- [29] *A highly spatially resolved ecosystem model for the North West European Continental Shelf*, J.I. Allen, J.C. Blackford, J.T. Holt, R. Proctor, M. Ashworth and J. Siddorn, *SARSIA* **86** (2001) pp. 423-440.

Acknowledgements

The success of the early stages of the HPCx project and of the implementation of the application benchmarks owes considerably to the efforts of many people. Among those without whom this work would not have been possible are Richard Blake, Paul Durham, Paul Sherwood, Bill Smith and Nic Harrison from CLRC, Mike Brown, Alan Simpson and Arthur Trew from EPCC, and Luigi Brochard and colleagues from IBM.

We are grateful to CSAR, ORNL, PSC and SARA for access to machines.