Future Proof Parallelism for Electron-Atom Scattering Codes on the XT4

Andrew Sunderland, Mike Ashworth, *Advanced Research Computing,*

Cliff Noble, Martin Plummer, *Atomic and Molecular Physics,*

Department of Computational Science and Engineering,

STFC Daresbury Laboratory, Warrington, United Kingdom

ABSTRACT: *Electron collisions with atoms were among the earliest problems studied using quantum mechanics. However, the accurate computation of much of the data required in astrophysics and plasma physics still presents huge computational challenges, even on the latest generation of high-performance computer architectures, such as the Cray XT series. In recent years a suite of parallel programs based on the 'R-matrix' ab initio approach to variational solution of the many-electron Schrodinger equation has been developed and has enabled much accurate scattering data to be produced. However, future calculations will require substantial increases in both the numbers of channels and scattering energies involved in the R-matrix propagations. This paper describes how these computational challenges are being addressed by improving the parallel performance of the PRMAT code on the Cray XT4.*

KEYWORDS: atomic physics, electron-atom scattering, R-matrix, parallel computing, optimization, multi-core systems

1. Introduction

Electron-atom and electron-ion scattering data are essential in the analysis of important physical phenomena in many scientific and technological areas. These include the development of environmentally safer alternatives to replace mercury vapour lighting, laser-produced plasmas as sources for next-generation nanolithography tools, the understanding of atmospheric processes, diagnostics of impurities in fusion plasmas and the quantitative interpretation of astrophysical data. In addition, the EU (and STFC-CLF) HiPER project (http://www.hiper-laser.org) for laser-ignited fusion, with associated experiments in laboratory astrophysics, opacity measurements, laser-excited hollow atoms and atoms in strong magnetic fields will stimulate detailed calculations of atomic scattering data. Despite the importance of these applications little accurate collision (theoretical or experimental) data is available for many complex atoms and ions. In particular this is the case for electron impact excitation and ionization at intermediate energies near the ionization threshold.

The R-matrix method [1] is an ab initio variational solution to the Schrödinger equation (and its relativistic extensions) for electron-atom (and molecule) scattering problems. Configuration space is partitioned by a sphere containing the target atom (molecule) or ion, outside of which the target wavefunctons are negligible and

exchange may be neglected. Inside the sphere an all-electron configuration interaction (CI) treatment produces a 'full' set of eigen-solutions for the system, independent of the scattering energy. The energy-dependent R-matrix is formed on the surface of the sphere from the eigenvalues and surface amplitudes of the inner region solutions. Outside the sphere many coupled differential equations must be solved for the scattering electron with potentials derived from the inner region CI target expansion. The R-matrix matches the inner and outer region solutions. In certain specialized versions of R-matrix theory for ionization two electrons are directly solved for outside the sphere: in the more standard R-matrix theory ionization is allowed for by the inclusion of square-integrable 'pseudostates' representing ionization channels, inside the sphere [1].

2. Description of software

2.1 The PRMAT Package

Over the last thirty years a suite of programs based upon the R-matrix approach has been developed and has enabled much accurate scattering data to be produced [1]. However, many problems of importance are not practical with programs designed to run serial computers, and a suite of parallel Fortran 95 codes for electron-atom scattering, PRMAT, funded by EPSRC, has been designed and implemented [2]. PRMAT is one of the application packages required to be provided on the UK's National Supercomputing Services HECToR [3] and HPCx [4], and consists of RMATRX2/95, based on the serial code RMATRX2 [5], and PFARM, based on the serial code FARM (Flexible Asymptotic R-matrix Package) [6]. RMATRX2/95 performs the inner region calculations. PFARM uses the results from RMATRX2/95 to form the energy-dependent R-matrix, then solves coupled differential equations over all scattering channels by propagating this matrix outwards from the sphere and matching the solutions to asymptotic boundary conditions, hence producing the required scattering data in both individual-atom and, after integration over energy, temperature dependent form. For complex atoms solutions are required for a dense set of scattering energies. The PRMAT package has been used to calculate data for electron collisions with various ions of Fe, Ni, Sn and neutral O. It is also being used for studies of intermediate energy scattering by light atoms [7].

Recently the package has been extended to include relativistic effects (needed for detailed treatment of open d-shell atoms and ions, for example) with the practical effect that the number of scattering channels in PFARM for systems of interest may now be much larger than for which the code was originally designed. This is also the case for intermediate energy scattering in which very large numbers of channels arise from a discretized electronic continuum inside the sphere. In addition, the complexity of the resonance structure for low energy electron scattering requires cross sections to be determined at typically tens of thousands of scattering energy values in order to yield accurate effective collision strengths.

2.2 PFARM

PFARM divides configuration space into radial sectors as shown in Fig. 1 and solves for the Green's function within each sector using a basis expansion. This approach is based upon the Baluja-Burke-Morgan (BBM) method [8].

In this implementation a variant of the BBM method is used to solve the coupled second-order differential equations defining the external region scattering. R-matrices at successively larger radial distances are obtained using Green's functions defined within finite radial sectors. The Green's functions are obtained using a shifted-Legendre basis.

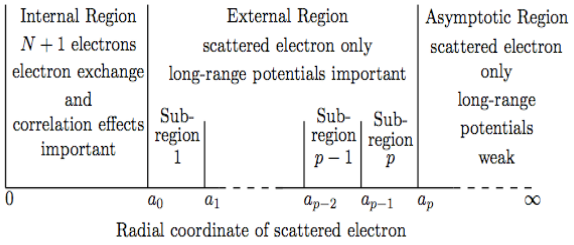

Figure 1. Partitioning of Configuration Space in PRMAT

PFARM takes advantage of standard highly optimized parallel numerical library routines and uses MPI for message-passing. The calculation proceeds in two distinct stages called **EXDIG** and **EXAS**.

In EXDIG, the Hamiltonian in each sector is generated and diagonalized in a distributed data parallelization, once only for all energies, using the ScaLAPACK library routine PDSYEVD. The sector Hamiltonians are diagonalized consecutively using all allocated processes.

The machine is then reconfigured with groups of processes assigned to specific tasks for EXAS, as shown in Fig. 2. The production group of processes RMPROD (R-matrix Production) calculates the initial R-matrix at the internal region boundary for successive scattering energies. The propagation group of processes RMPROP forms a systolic pipe along which the stream of R-matrices is passed as they are propagated from the internal region boundary to the external region boundary. The pipeline unit is replicated across the machine: three such pipelines are shown in Fig. 2. Asynchronous non-blocking messages are used to pass R-matrices between the nodes in the propagation pipeline. At the end of the pipeline the asymptotic group of processes RMASY (Asymptotic Region Calculation) calculates the scattering K-matrix and collision strengths for transitions between all the states included in the R-matrix expansion for the current scattering energy. Finally a single process is dedicated to gathering the results from the asymptotic group. Results are stored to disk periodically to provide a restart procedure, as runs involving many thousands of scattering energies may take considerable time.

Significant savings in computation time and memory requirements in EXAS are also obtained by taking advantage of decoupling of channels in the external region, partitioning them into two non-interacting groups according to target spin (non-relativistic calculations) or to the intermediate-coupling K quantum number (for relativistic calculations). In the most favourable cases this has the effect of splitting the problem into two roughly equal sized parts. For each block, separate pipelines exist, though the propagation of the block-split R-matrices across sectors is coupled and some message passing between processes is required for each equivalent sector calculation. This decoupling is represented in Fig. 2 by the two rows of processes within a pipeline.

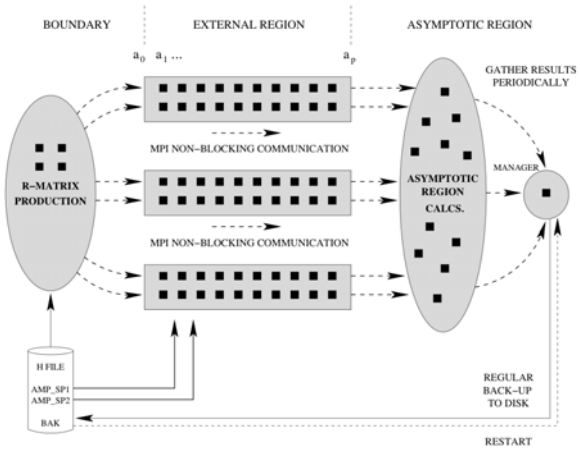

Figure 2. Assignment of tasks to sub-groups of processes in the **EXAS** stage of PFARM

A further reduction in compute time can usually be obtained by undertaking EXAS runs in two stages. Firstly a *fine region propagation* involving scattering energies residing in the extremely complex scattering resonance region followed by a *coarse region propagation* for scattering energies above this region. This partitioning allows us to optimize sector length for each region: generally smaller sector length with a larger number of basis functions in the fine region and a larger sector length with a smaller number of basis functions in the more benign coarse region. In most cases the vast majority of energy points lie within the fine region and therefore this is the stage where most compute time is spent.

The particular advantages of the parallelization are:

- The costly parallel diagonalization of large sub-region matrices is computed independently of scattering energies. This is particularly suitable for calculations involving hundreds or thousands of scattering energies.
- The sub-region propagation calculations make much use of Level 3 serial BLAS matrix-multiply routines. The highly optimized BLAS library routines are typically designed to take advantage of the underlying microprocessor architecture, and therefore attain near peak performance. Fig. 3, taken from a single-node efficiency study on HPCx [9], shows that PFARM is one of the fastest application codes,

averaging >35% of peak performance, or 2.1 Gflops/s.

- On most high-end computing platforms asynchronous non-blocking communication reduces communication costs.
- The approach scales well *provided* computational load is adequately balanced across functional groups. The number of processes assigned to functional groups RMPROD and RMASY must ensure that i) initial R-matrices are generated with sufficient frequency to maintain a fully operational pipeline; ii) collision strength calculations are processed at a sufficient rate to prevent bottlenecks.

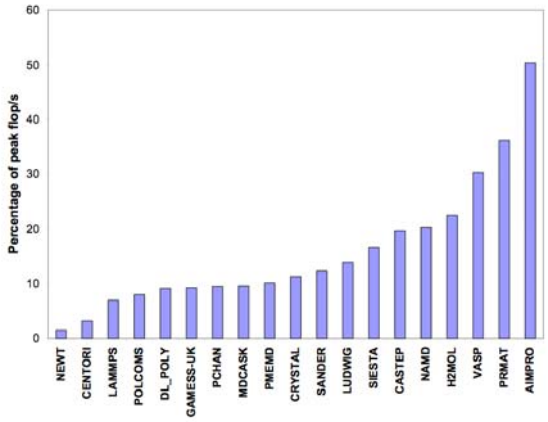

Figure 3. Single Node Performance of PFARM (labelled here as PRMAT) on HPCx

2.3 HECToR distributed-CSE projects

In 2008 the PRMAT package developers were awarded a contract for *Distributed Computational Science and Engineering (dCSE)* on HECToR, funded by NAG (Numerical Algorithms Group) Ltd on behalf of the UK Engineering and Physical Sciences Research Council (EPSRC) [9]. The aim of dCSE projects is to enable computational specialists to:

- port their codes onto HECToR, in particular to work with new codes or to enable previously unsupported features in existing codes;
- improve the performance and scaling (ideally to thousands of cores) of their codes on HECToR;
- re-factor their codes to improve long-term maintainability;
- take advantage of algorithmic improvements in the field of high-performance computing.

The award is to be used entirely for software development in order to increase code performance and/or utility, and hence to deliver further science. In the case of PRMAT the project has two main aims: optimization and enhanced paralellization of PFARM, plus development and incorporation of a new code using an Arnoldi log-derivative (ALD) propagator [10]. The ALD code implies a significant improvement in memory efficiency and, once fully incorporated, performance improvement. In this report we concentrate on the upgrading of PFARM: the code improvements described in the following sections have been undertaken as part of this project.

3. Description of Target Hardware

HECToR

HECToR [3] is the UK's latest National Supercomputing Service, located at the University of Edinburgh and run by the HPCx consortium. The HECToR Phase 1 Cray XT4 system comprises 1416 compute blades, each of which has 4 dual-core processor sockets. This amounts to a total of 11,328 cores, each of which acts as a single CPU. The processor is an AMD 2.8 GHz Opteron. Each dual-core socket shares 6 GB of memory, giving a total of 33.2 TB in all. The theoretical peak performance of the system is 59 Tflops/s, positioning the system at No. 46 in the November 2008 Top 500 list [12].

HPCx

For comparison purposes we will also present results from HPCx [4], the previous UK National Capability Computing service, located at the Computational Science and Engineering Department at STFC Daresbury Laboratory [12] and comprising of 160 IBM eServer 575 nodes. Each eServer node contains 16 1.5 GHz POWER5 processors, giving a total of 2536 processors for the system. The total main memory of 32 GBytes per frame is shared between the 16 processors of the frame. The frames in the HPCx system are connected via IBM's

High Performance Switch. The current configuration has a theoretical peak performance of 15.4 Tflops and is positioned at No. 490 in the November 2008 Top 500 list.

4. Initial performance of PFARM on the XT4

An Initial Performance analysis of PFARM was undertaken on HECToR using an FeIII test case dataset with jj-coupling (ie taking into account relativistic effects). This dataset is representative of recent work by users of PFARM. The propagation involves 1181 channels (equivalent to the dimension of the R-matrix), 10677 scattering energies in the fine region and 205 scattering energies in the coarse region. The version of PFARM used is compiled with the *–fast* option of the PGI compiler on the XT4 but otherwise is not optimized specifically for the XT4.

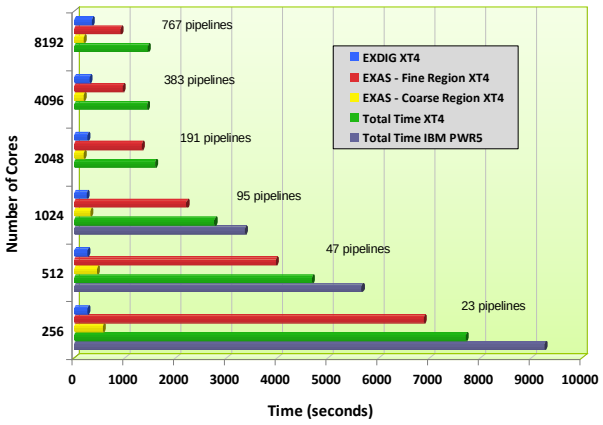

Figure 4. Initial parallel performance of PFARM on HECToR & HPCx

The timings for the various core counts on XT4 are broken down into the three stages - EXDIG, EXAS Fine Region and EXAS Coarse Region. Summing these times gives the total time taken for the complete external region calculation for this test case. For comparison, total time taken on the HPCx machine (IBM PWR5 p5-575) is included. The number of pipelines built for the fine region calculation is also shown in Fig. 4. This number rises almost linearly as the number of cores is increased as more pipelines are fitted into the arrangement shown in Fig. 2. It is shown that the fine region propagation scales well up to 2048 cores. This stage usually dominates overall compute time and thus total time also scales well up to 2048 cores. The code continues to speed-up when run on 4096 cores, but no further gains are made by running on 8192 cores. The speed of the coarse region propagation does not improve when running on more than 2048 cores. This is due to the maximum number of pipelines (68) having been reached for the limited number of scattering energies (205) in the coarse region.

5. Optimization of PFARM on the XT4

Fig. 4 shows that the EXDIG stage of the program is not scaling well on the XT4. Although this behaviour has little impact on overall time on lower core counts, EXDIG is accounting for around 24% of total run time on 8192 cores. Previous investigations [2] [12] have shown that parallel diagonalization is often a computational bottleneck in large-scale calculations and that parallel scaling is often limited even when using optimized numerical library routines.

5.1 Initial Analysis of Parallel Diagonalization Routines on the XT4

Several parallel eigensolver routines for solving standard and generalized dense symmetric or dense Hermitian problems are available in the current release of ScaLAPACK [14]. Previous investigations [15] on HPCx had found that the divide-and-conquer-based routine PDSYEVD [16] was most suited to parallel diagonalizations in PFARM, where all eigenpairs of the system are required.

Recently a new routine PDSYEVR [17], based on a Multiple Relatively Robust Representation approach, has been implemented by ScaLAPACK developers. This routine has been made available to users for early testing.

The relative performance of PDSYEVD and PDSYEVR on the XT4 and IBM p5-575 machines is shown in Fig. 5. The performance on lower core counts is roughly equivalent. However the established PDSYEVD routine performs much better on higher core counts and it was decided to continue with this routine at present. It should be noted that the PDSYEVR routine tested here is under development and the official release is likely to have improved performance. If these scaling issues are resolved then PDSYEVR is likely to be an attractive future option as this method promises much-reduced memory overheads compared to PDSYEVD, allowing larger cases to be solved on smaller core counts.

The performance of PDSYEVD on Hamiltonian sector matrix sizes ranging from N=20064 to N=62304 is shown in Fig. 6. As expected larger problem sizes scale much

better as these cases are characterised by a more favourable communication/computation ratio.

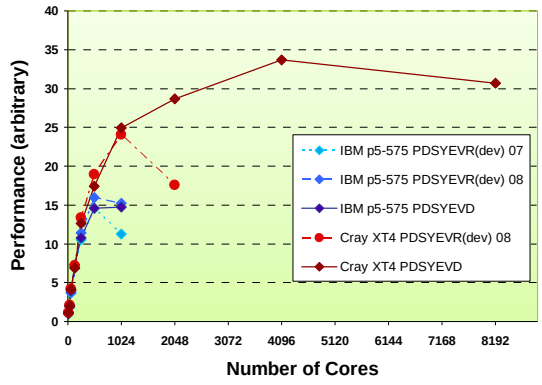

Figure 5. ScaLAPACK Parallel Diagonalizer Performance on HPCx (IBM p5-575) and HECToR (XT4) for FeIII case with Hamiltonian dimension N = 220064.

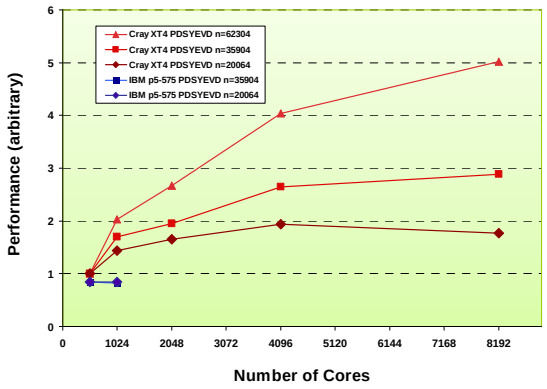

Figure 6. PDSYEVD Performance on HECToR (XT4) and HPCx (IBM p5-575) and for FeIII cases with a range of Hamiltonian dimensions.

5.2 Optimization of EXDIG on the XT4

The original EXDIG code steps through each sector sequentially, undertaking a parallel diagonalization on each sector Hamiltonian matrix. In order to improve the parallel performance of EXDIG the code has been modified to calculate each Hamiltonian sector parallel diagonalization concurrently on sub-groups of processes. These sub-groups are created by partitioning the global Blacs-based grid. The sub-groups need to be of sufficient size to accommodate the Hamiltonian sector matrix plus associated overheads but small enough to maintain an advantageous communications/computation ratio. Overall this strategy avoids distributing computational loads too thinly when problem sizes are relatively small and parallel jobs involve thousands of cores, i.e. it shifts the parallel scaling behaviour towards the steeper gradients associated with the lower-to-medium core counts in Fig. 5 and Fig. 6. It should be noted that this new approach involves extra distribution and set-up costs.

The performance improvements on the XT4 for an FeIII case is shown in Fig. 7. In the new version of the code the four Hamiltonian sector matrices are distributed to four sub-groups of cores of size NP / 4, where NP is the total number of cores. In this case, using the optimized EXDIG code results in a speed-up of 2.87 on 8192 cores of the XT4. The EXDIG modifications described here will ensure that ambitious calculations planned by PFARM users, with more complex and longer-range potentials and thus involving larger sector matrices and more sectors, will scale well to much larger numbers of cores than this example.

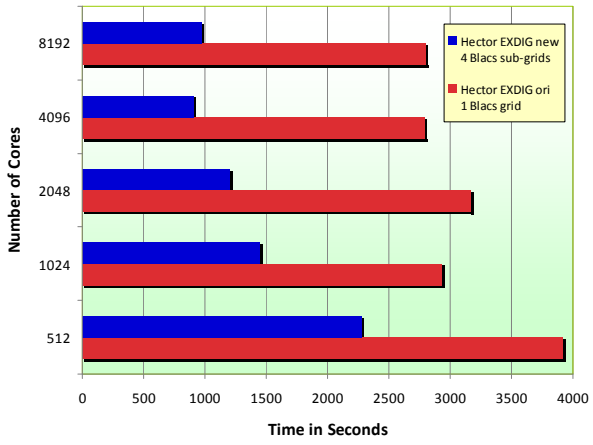

Figure 7. Optimized parallel diagonalization performance on HECToR Cray XT4, Sector Hamiltonian Dimension = 44878.

5.3 Optimization of EXAS on the XT4

Appropriate allocation of processes to tasks in EXAS is critical for efficient parallel performance. Most importantly, the RMPROD and RMASY groups at the start and ends of pipelines must be of sufficient size to i) generate initial R-matrices with sufficient frequency to maintain a fully operational pipeline; ii) process collision strength calculations at a sufficient rate to prevent bottlenecks at the end of pipelines.

A load-balancing analysis was previously undertaken on the Cray T3E when the PFARM code was initially developed. Computational experiments that varied the number of processes in the task groups were reported in publications such as [1]. These now needed to be updated for the XT4 in order to reflect modern multi-core MPP architectures. Figure 8 summarises a series of test runs undertaken on the XT4 with 1024 cores for an energy-reduced fine region propagation. The number of cores in the RMPROD pool is varied from 224 to 320 and the number of cores in the asymptotic task group *per* pipeline is varied from 2 to 9. It is shown that the ideally balanced configuration for this case is 288 cores in RMPROD and 7 cores in RMASY per pipeline. This optimized configuration results in a 28% reduction in run-time when compared to the worst load-balanced configuration (320 cores in RMPROD and 2 cores per pipeline in RMASY).

Applying this load-balancing configuration to the initial full fine region propagation runs from Fig. 4 results in the performance improvements shown in Fig. 9. The improved load-balancing continues to have a substantial impact on speed on 2048 cores, but the gains are negligible on 4096 cores and 8192 cores. For these cases further analysis will be necessary, but it is believed that the single manager process is now the bottleneck. Replacing this with a new task group where multiple manager processes can be specified will help alleviate these imbalances and is a priority task for the ongoing work of the project.

Currently users must specify the allocation of cores to task groups in their input files. Short scripts have been written that will help determine process allocations automatically, based upon the problem characteristics and the underlying hardware, and these are currently being updated for the HECToR machine.

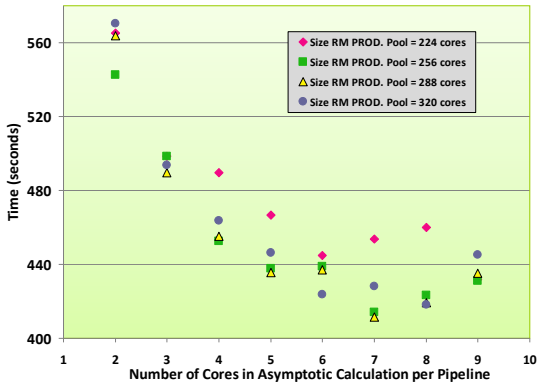

Figure 8. Load-balancing analysis on Hector Cray XT4

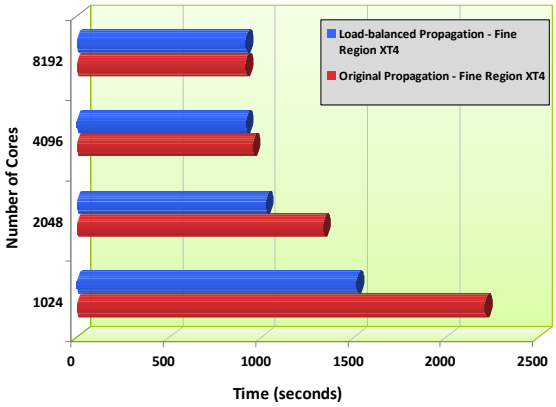

Figure 9. The effect on performance of load-balancing EXAS for the fine region propagation

6. Future work on PFARM and the ALD code

In addition to ongoing tasks indicated in the previous section, further potential optimizations to PFARM have been identified and it is expected that many of these will be carried out over the remainder of the project.

Further improvements include:

1. Greater flexibility in the assignment of sectors to processes. The code was originally designed for architectures with memory resource per process a small fraction of that available on HECToR. This

resulted in a programming model where each pipeline process is always designated one spin-split sector calculation regardless of problem size and memory availability. With target calculations now increasing by up to an order of magnitude, this rigid mapping needs to be relaxed in order to permit:

a) More than one process per sector calculation (for larger jobs), achieved via a data distribution parallelization of the R-matrix sector propagation, i.e. pipeline *nodes* compromising of groups of processes.
b) More than one sector per process (for smaller jobs) in order to fully utilize local memory resources and reduce communication overheads.
c) More flexibility in parallelizing channel splitting within a pipeline to allow for differing sizes of sub-matrices.

2. An assignment of tasks to processes that reduces communications between the underlying hardware. For example on HECToR, neighbouring sectors should reside within a multi-core processing element whenever possible.

3. The introduction of task-harnessing to calculate different groups of partial waves and energies concurrently. The process arrangement shown in Fig. 2 would be replicated across HECToR resources for each, or groups of, partial waves (scattering symmetries) and energies associated with the calculation.

4. Parallel I/O features will be introduced into the code where appropriate.

The HECToR service is soon to be upgraded to Phase 2a. This will involve a change from a dual-core to a quad-core architecture. A consequence of the technology refresh is that memory per core will be reduced from 3GBytes per core to 2 GBytes per core. However memory accesses are expected to be somewhat more efficient on the Phase 2a machine. Further optimizations carried out on PFARM will reflect these changes to the underlying architecture.

A substantial part of the dCSE project is dedicated to introducing the ALD propagator into PFARM. ALD is a stable propagation method for solving coupled sets of Schrödinger equations introduced by Alexander and Manolopoulos [10]. Within each sector the potential coupling the equations is approximated by a linear reference potential. An optimum reference potential is obtained by diagonalizing the full potential coupling matrix at two points defined by a Gauss integration mesh scaled to the radial sector. Exact solutions of the sector equations with a linear reference potential are given by Airy functions. These may be computed accurately and efficiently. The size of matrices associated with Airy LD propagations is much reduced in comparison with the BBM approach, at the expense of needing more sectors, and ALD is expected to maintain better performance at capability usage for larger problems.

So far, the method has been introduced into a new object-oriented (within Fortran 2003 limits) version of FARM, which also incorporates the mixture of symmetry-based block-partitioned and block-diagonal matrices directly into the propagation procedure via appropriate use of derived datatypes. The parallelization of this code includes advantageous use of passive communication features and an efficient customized parallel I/O library. The latter has already been incorporated into the main EXDIG code and is currently being installed in EXAS.

7. Conclusions

In this paper we have described some of the optimizations being undertaken on the parallel R-matrix program PFARM for runs using the Cray XT4. The code combines a unique, highly flexible functional and data decomposition approach for external region R-matrix propagation. The code is built on highly optimized parallel numerical library routines and is therefore highly efficient. However modifications to the code need to be made in order to maximise performance on the new generation of high-end computing resources, typically with many thousands of multi-core processors. Improvements to parallel scaling performance, single-node efficiency and memory usage will enable very large electron-atom and electron-ion scattering calculations to be addressed on these machines. The PFARM code may also be reasonably straightforwardly adapted for outer region electron-molecule scattering. We have demonstrated that code optimizations undertaken to-date have already yielded significant increases in performance of the code on the XT4.

Acknowledgments

The authors would like to thank the following colleagues and collaborators for their contributions to this work:

Val Burke, *STFC Daresbury Laboratory*

Phil Burke, *Queen's University Belfast*

Christof Voemel, *ETH Zurich.*

About the Authors

Dr Andrew Sunderland (andrew.sunderland@stfc.ac.uk) is a computational scientist in the Advance Research Computing Group, Computational Science and Engineering, at STFC Daresbury Laboratory. His work involves the development, optimization and analysis of scientific codes and parallel numerical algorithms.

Dr Mike Ashworth (mike.ashworth@stfc.ac.uk) is Group Leader for the Advanced Research Computing Group, Computational Science and Engineering, at STFC Daresbury Laboratory. His own work focuses on the development and optimisation of environmental modelling and CFD codes, including performance engineering and application of Grid technologies.

Professor Cliff Noble (cliff.noble@stfc.ac.uk) is a visiting professor at the Department of Computational Science and Engineering at STFC Daresbury Laboratory. His previous role was Head of the Atomic and Molecular Physics Group in the department. The focus of his work is developing codes that calculate atomic and molecular collision processes using high performance computers.

Dr Martin Plummer (martin.plummer@stfc.ac.uk) is a computational scientist and Group Leader for Theoretical Atomic and Molecular Physics in the Advanced Research Computing Group, Computational Science and Engineering, at STFC Daresbury Laboratory. He also optimizes and manages various codes on the UK HPC service HPCx particular responsibility for the materials science code CASTEP.

All the above authors can be contacted at:

STFC Daresbury Laboratory,
Warrington,
WA4 4AD
United Kingdom

Tel: +44 1925 603000

References

1. P G Burke, C J Noble and V M Burke, Adv. At. Mol. Opt. Phys. 54 (2007) 237-318.
2. A G Sunderland, C J Noble, V M Burke and P G Burke, Computer Physics Communications (CPC) 145 (2002), 311-340.
3. HECToR – UK National Supercomputing Service, http://www.hector.ac.uk.
4. HPCx – The UK's World-Class Service for World-Class Research, http://www.hpcx.ac.uk.
5. P G Burke, V M Burke and K M Dunseath, J Phys. B 21 (1994), 5341-5373.
6. V M Burke and C J Noble, CPC 85 (1995), 471-500.
7. BM McLaughlin, MP Scott, AG Sunderland, CJ Noble, VM Burke, CA Ramsbottom, R.H.G. Reid, A. Hibbert, K.L. Bell, P.G. Burke, Atomic Data Nucl Data Tables 93, 55-104 (2007); M Lysaght, PhD thesis, University College Dublin, Ireland (2006); MP Scott, CA Ramsbottom, CJ Noble, VM Burke, PG Burke, J Phys B: At. Mol. Opt. Phys. 39, 387-400 (2006); BM McLaughlin, A Hibbert, MP Scott, CJ Noble, VM Burke, PG Burke, J Phys B: At. Mol. Opt. Phys. 38, 2029-2045 (2005); M Plummer, CJ Noble, M Le Dourneuf J Phys B: At. Mol. Opt. Phys. 37, 2979-2996 (2004); see also 'Mathematical and computational methods in R-matrix theory', eds. M Plummer, J D Gorfinkiel and J Tennyson, CCP2, STFC Daresbury Laboratory, UK, 2007 (on line version http://www.ccp2.ac.uk).
8. K L Baluja, P G Burke and L A Morgan, CPC 27 (1982), 299-307.
9. Single Node Performance Analysis of Applications on HPCx, M. Bull, HPCx Technical Report HPCxTR0703 2007, http://www.hpcx.ac.uk/research/hpc/technical_reports/HPCxTR0703.pdf.
10. Alexander and Manolopoulos, J. Chem. Phys. 86 (1987), 2044-2050).
11. HECToR – Distributed CSE Support, http://www.hector.ac.uk/cse/distributedcse.
12. STFC's Computational Science and Engineering Department, http://www.cse.scitech.ac.uk/.

13. TOP 500 Supercomputing Sites, http://www.top500.org.
14. The ScaLAPACK Project, http://www.netlib.org/scalapack/index.html.
15. Using Scalable Eigensolvers on HPCx: A Case Study, Ian Bush, Andrew Sunderland, Gavin Pringle, HPCx Technical Report HPCxTR0510 2005, http://www.hpcx.ac.uk/research/hpc/technical_reports/HPCxTR0705.pdf.
16. A Parallel Divide and Conquer Algorithm for the Symmetric Eigenvalue problem on distributed memory architectures, F.Tisseur and Jack Dongarra, SIAM J. SCI. COMPUT, Vol.20, No. 6, pp. 2223-2236 (1999).
17. PDSYEVR. ScaLAPACK's parallel MRRR algorithm for the symmetric eigenvalue problem}, D.Antonelli, C.Vomel, Lapack working note 168, (2005), http://www.netlib.org/lapack/lawnspdf/lawn168.pdf.