# Parallel Eigensolver Performance on High Performance Computers[1]

Andrew Sunderland
STFC Daresbury Laboratory, Warrington, UK

**Abstract**

Eigenvalue and eigenvector computations arise in a wide range of scientific and engineering applications and usually represent a huge computational challenge. It is therefore imperative that suitable, fast and scalable parallel eigensolver methods are used in order to facilitate the efficient solution of the most demanding scientific problems. This paper analyzes and compares the performance of some of the latest eigensolver algorithms, including a pre-release ScaLAPACK routine, on contemporary high-end systems such as the 11,328 core Cray XT4 system HECToR in the UK. The analysis involves symmetric matrix examples obtained from current problems of interest for two large-scale scientific applications.

## 1 Introduction

Efficient parallel eigensolver performance is essential for many parallel scientific and engineering application codes. For example, in quantum chemistry and quantum physics the computation of eigenvalues may be required in order to calculate electronic energy states. Computations often involve matrices of dimension of tens or even hundreds of thousands that need to be solved quickly with manageable memory requirements on the latest large-scale high-performance computing platforms. This paper analyses the performance of parallel eigensolver library routines across a range of applications, problem sizes and architectures. New developments of particular note include a pre-release ScaLAPACK implementation of the Multiple Relatively Robust Representations (MRRR) algorithm and the next generation series of high end parallel computers such as the Cray XT series and IBM's BlueGene. The results presented are based upon Hamiltonian matrices generated during electron-atom scattering calculations using the PRMAT code [1] and matrices from the CRYSTAL [2] package generated during the computation of electronic structure using Hartree-Fock theory.

[1] Presented at the Cray User Group Conference, Helsinki, Finland, May 2008

# 2 Parallel Eigensolver methods

## *2.1 The Symmetric Eigensolver Problem*

The standard eigenvalue problem is described as

$$\mathbf{A}\,\mathbf{x} = \lambda\,\mathbf{x}$$

where **A** is a matrix and λ is the eigenvalue corresponding to eigenvector **x**.

For symmetric matrices this equation can be rearranged to give the equation describing the diagonalization of matrix **A**:

$$\mathbf{A} = \mathbf{Q}\,\Lambda\,\mathbf{Q}^{\mathbf{T}}$$

where the columns of the matrix **Q** are represented by the orthogonal eigenvectors of **A** and the diagonal matrix $\Lambda$ represents its associated eigenvalues.

## *2.2 Underlying equations for matrix diagonalizations in PRMAT*

The PRMAT code is based on the Baluja-Burke-Morgan [3] approach for solving the non-relativistic Schrodinger equation describing the scattering of an electron by an N-electron atom or ion:

$$\mathbf{H}_{N+1}\,\Psi = \mathbf{E}\,\Psi$$

where **E** is the total energy in atomic units and $\mathbf{H}_{N+1}$ is the (N+1)-electron Hamiltonian matrix. In this approach the representative of the Green's function $\mathbf{(H + L - EI)^{-1}}$ is diagonalized within a basis. The symmetric matrix **(H + L - EI)** is reduced to diagonal form by the orthogonal transformation:

$$\mathbf{X}^{\mathbf{T}}\,\mathbf{(H + L - E)}\,\mathbf{X} = \mathbf{(E_k - E)}$$

where the columns of the orthogonal matrix $X^T$ represent the eigenvectors and $E_k$ the eigenvalues of (H + L).

## 2.3 Underlying algorithms for matrix diagonalizations in CRYSTAL

The CRYSTAL package [2] performs ab initio calculations of the ground state energy, electronic wave function and properties of periodic systems. Development of the software has taken place jointly by the Theoretical Chemistry Group at the University of Torino and the Computational Materials Science Group at STFC Daresbury Laboratory (UK) [14]. The computation of the electronic structure is performed using either Hartree-Fock or Density Functional theory. In each case the fundamental approximation made is the expansion of the single particle wave functions as a linear combination of atom centred atomic orbitals (LCAO) based on Gaussian functions.

## 2.4 Review of Symmetric Eigensolver Methods

The solution to the real or hermitian dense symmetric eigensolver problem usually takes place via three main steps

1. *Reduction of the matrix to tri-diagonal form*, typically using Householder Reduction,
2. *Solution of the real symmetric tri-diagonal eigenproblem* via one of the following methods:
    - Bisection for the eigenvalues and inverse iteration for the eigenvectors [4],[5],
    - QR algorithm [6],
    - Divide & Conquer method (D&C) [7] ,
    - Multiple Relatively Robust Representations (MRRR algorithm) [8],
3. *Back transformation* to find the eigenvectors for the full problem from the eigenvectors of the tridiagonal problem.

For an $n \times n$ matrix, the reduction and back transformation phases each require $O(n^3)$ arithmetic operations. Until recently, all algorithms for the symmetric tridiagonal eigenproblem also required $O(n^3)$ operations in the worst case and associated memory overheads of $O(n^2)$. However, for matrices with clustered eigenvalues, the Divide and Conquer method can take advantage of a process known as *deflation* [7], which often results in a reduced operation count. The potential advantages of the MRRR algorithm are twofold in that theoretically only $O(kn)$ operations are required, where $k$ is the number of desired eigenpairs, and the associated memory requirements are only $O(n)$.

### 2.5 Parallel Library Routines for Solving Dense Symmetric Eigenproblems

Several eigensolver routines for solving standard and generalized dense symmetric or dense Hermitian problems are available in the current release of ScaLAPACK [9].

These include:

- *PDSYEV* based on the QR Method
- *PDSYEVX* based on Bisection and Inverse Iteration
- *PDSYEVD* based on the Divide and Conquer method
- Also tested here is a new routine *PDSYEVR* [10] based on the MRRR algorithm. At the time of this analysis this routine is a pre-release version and is still undergoing testing and development by ScaLAPACK developers.

*PDSYEV* and *PDSYEVD* only calculate *all* the eigenpairs of a matrix. However both *PDSYEVX* and the new *PDSYEVR* have the functionality to calculate subsets of eigenpairs specified by the user. For conciseness the performance results reported in this paper focuses on the latest parallel solvers *PDSYEVD* and *PDSYEVR*.

## 3 Testing Environment

### 3.1 Test Matrices

The matrices analysed here generated by the R-matrix package are derived from external sector Hamiltonian Ni3+ and Fe+ scattering calculations using the PRMAT code. They are all real symmetric dense matrices with dimensions ranging from 10032 to 62304. The eigenvalue distribution is fairly well-spaced with comparatively few degeneracies, though some clustering does exist. For other cross-platform comparisons, diagonalizations using matrices obtained from the CRYSTAL package have been measured, where the matrix sizes range from a dimension of 7194 to 20480. The eigenvalue distribution of these real symmetric matrices is typically much more clustered than those obtained from the PRMAT code.

### 3.2 High Performance Computing Platforms

Parallel timings presented are mainly from runs undertaken on two of the UK's National Supercomputing facilities run by the HPCX Ltd consortium: the new *HECToR* Cray XT4 machine [12] with 11,328 AMD 2.8 GHz Opteron cores sited at the University of Edinburgh and *HPCx* [13] at STFC Daresbury Laboratory [14] comprising of 160 IBM p5-575 nodes, totalling 2536 processors

Some figures also demonstrate the evolution of the HPCx system over the past five years. The original Phase 1 configuration consisted of p690 processors with the

colony (SP) switch. Figures also show timing comparisons of runs taken on HPCx with runs undertaken on other contemporary HPC platforms: a 2048 core IBM Blue Gene/L and 4096 core IBM Blue Gene/P machine [15], also sited at STFC Laboratory and a Cray XT3 machine sited at the Swiss Supercomputing Centre CSCS [16] with AMD 2.6 GHz Opteron processors.

The results include comparisons for dual-core processors (Cray XT4, BG/L), quad-core processors (BG/P) and 16-way and 32-way shared-memory processors (SMPs) (IBM p5-575 and IBM p690). For reasons of consistency, throughout the performance analysis charts 'Number of Processors' is equivalent to 'Number of Cores'.

## 4 Performance Results

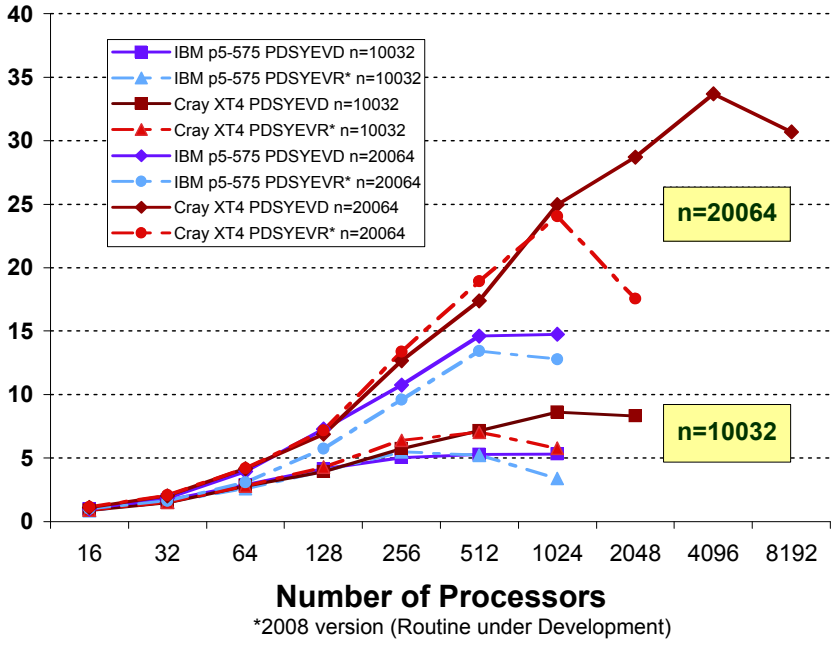


Figure 1. Relative Scaling of *PDSYEVD* and *PDSYEVR* for Hamiltonian matrices from PRMAT

Figure 1 shows the parallel scaling of performance for the diagonalization routines *PDSYEVR* and *PDSYEVD* for a range of problem sizes from PRMAT on the current configuration of the Cray XT4 and IBM p5-575. The relative performance reported is the time taken to solution for the algorithm and HPC platform with respect to the CPU time of *PDSYEVD* on 16 processors of HPCx. The performance results for two different versions of *PDSYEVR* are shown: an older version from 2007 and a more recent version from early 2008. It is shown that the parallel performance of *PDSYEVR* (2008) is very close to that of *PDSYEVD* for the two problem sizes, though the performance on the processor counts of around one thousand can

degrade, possibly due to uneven distributions of the eigenvalue representation tree amongst processors [17].

Figure 1 also shows how performance increases up to a maximum of 4096 processors on the Cray XT4 for the larger problem size. Performance is around 34 times faster on 4096 processors than on 16 processors. Ideal parallel scaling here would result in a performance improvement factor of 256 (4096/16).

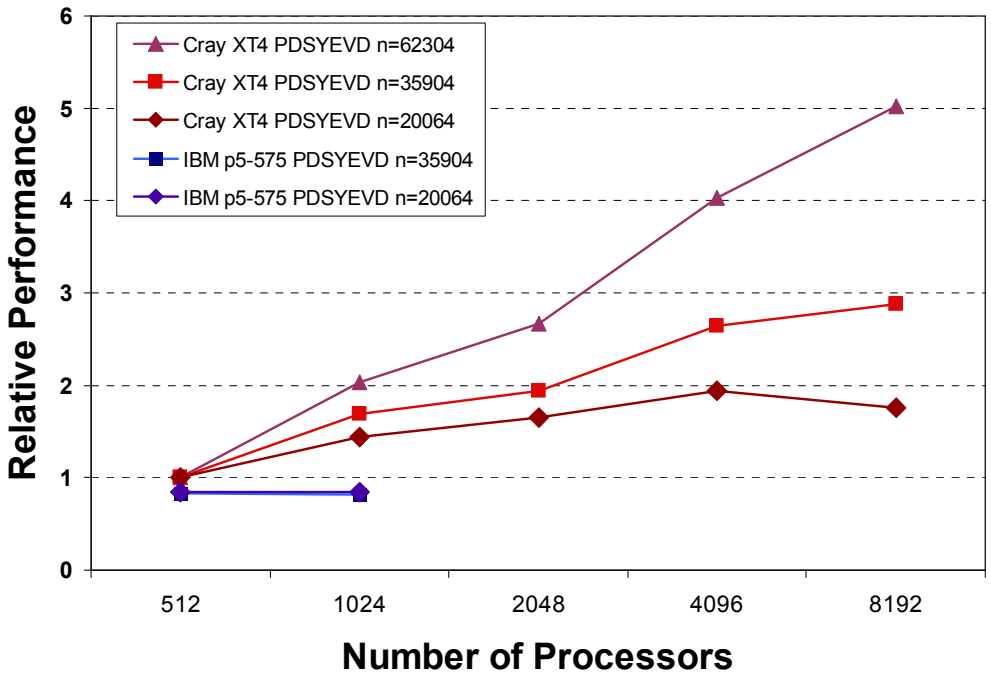


Figure 2. Parallel Performance of *PDSYEVD* for large matrix sizes on the Cray XT4 and IBM p5-575

Figure 2 compares the performance of *PDSYEVD* on processor counts upwards of 512 on the Cray XT4 and IBM p5-575. Relative performance for the three matrices is based on a 512 processor run on the Cray XT4. The eigensolution for three Hamiltonian matrices from PRMAT are considered up to a maximum matrix dimension of 62304. It can be seen that the parallel scaling on the Cray XT4 is significantly better than on the IBM p5-575 at these high processor counts. The parallel performance of *PDSYEVD* for the largest problem size here is good right out to the maximum job size (8192 processors) on Hector.

0 and Figure 4 show how the *PDSYEVD* routine scales with processor count on the high-end computing platforms detailed in section 3.2 for CRYSTAL matrices. Parallel performance is best on the Cray XT machines for both matrices tested here, relatively closely matched by the current configuration of HPCx (IBM p5-575 with the High Performance Switch).

At the time these tests were undertaken the Cray XT3 testing platform consisted of single core processors only. Here the higher clock speed of the dual core XT4 processors relative to the XT3 results in a negligible improvement to parallel performance. The performance of the IBM BlueGene/L and BlueGene/P is around three times slower than the Cray XTs, roughly matching the performance of the orginal HPCx system (p690 SP).

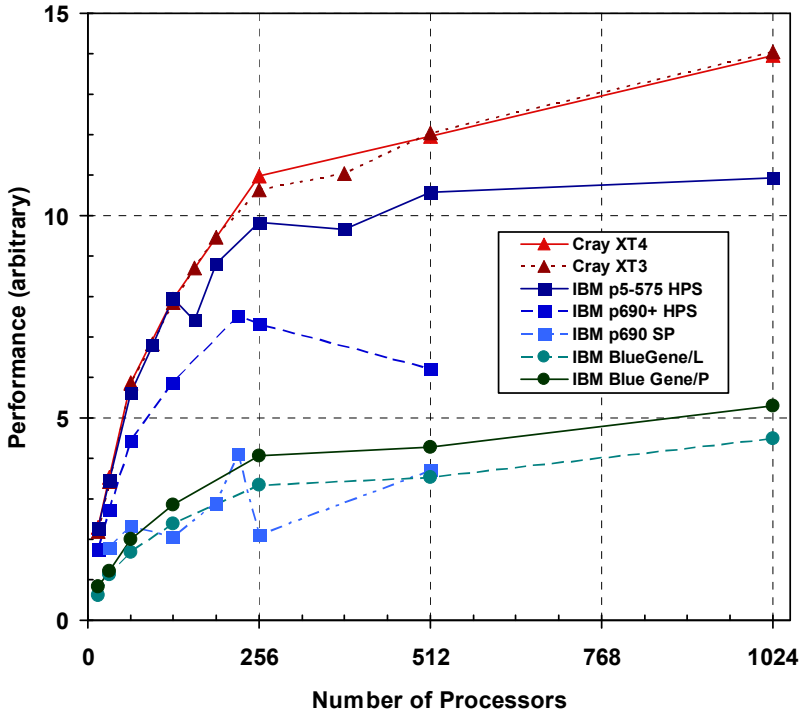


Figure 3. Performance of PDSYEVD on current HPC architectures (CRYSTAL matrix, n=7194).

The advantage of the BlueGene/P machine is demonstrated most clearly in Figure 5, where the estimated power consumption for parallel matrix eigensolves relative to that undertaken on 16 processors of BlueGene/P is shown. These figures are based on the Mflops/Watt values given on the Green Top 500 listing [18] . In this alternative listing the BlueGene/P at STFC Daresbury Laboratory in the United Kingdom is ranked as number 1 i.e. it is currently the most efficent supercomputer (with respect to flops per Watt) in the world.

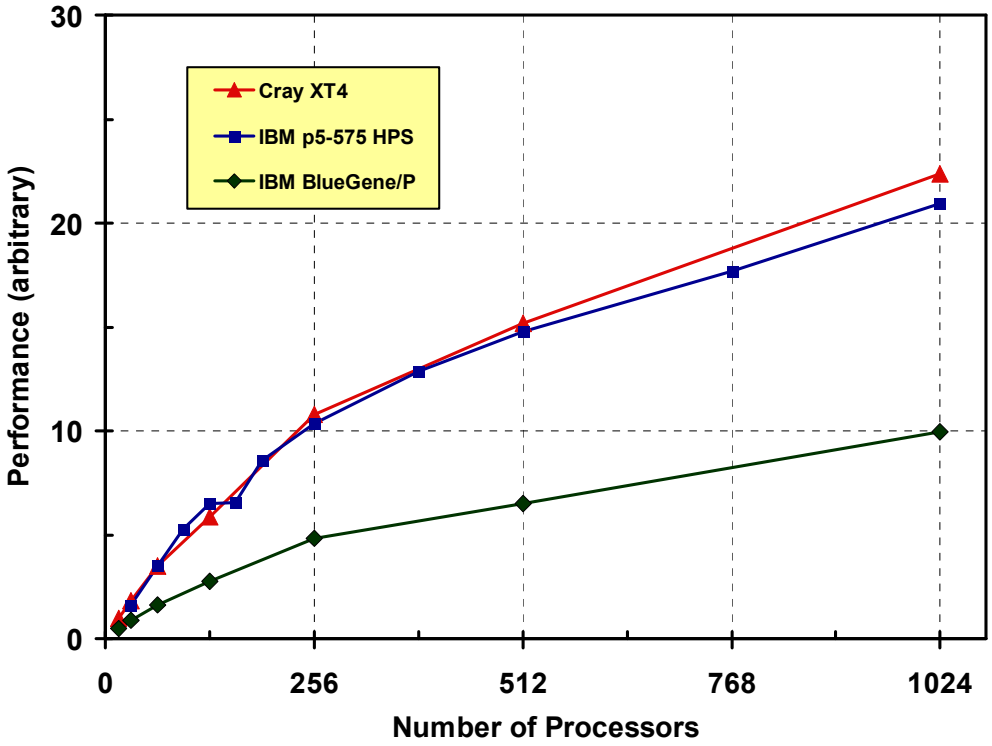


Figure 4. Performance of *PDSYEVD* on current HPC architectures (CRYSTAL matrix, n=20480).

One characteristic of new parallel eigensolvers that has become evident during the course of the tests is that the tridiagonal eigensolver is no longer the primary computational bottleneck during the full symmetric eigensolve. Figure 6 shows how the balance between reduction, tri-diagonal eigensolve and back transformation changes significantly with different eigensolver methods. It can be seen that the time taken in the tri-diagonal eigensolve using Divide-and-Conquer is now relatively small compared to the time taken in reducing the full matrix to tri-diagonal form. Although the back transformation calculation scales very well to large numbers of processors, the relative computational costs of the reduction phase remain high. This contrasts markedly with the traditional QR-based approach where the tri-diagonal eigensolve dominates the overall time taken to solution.

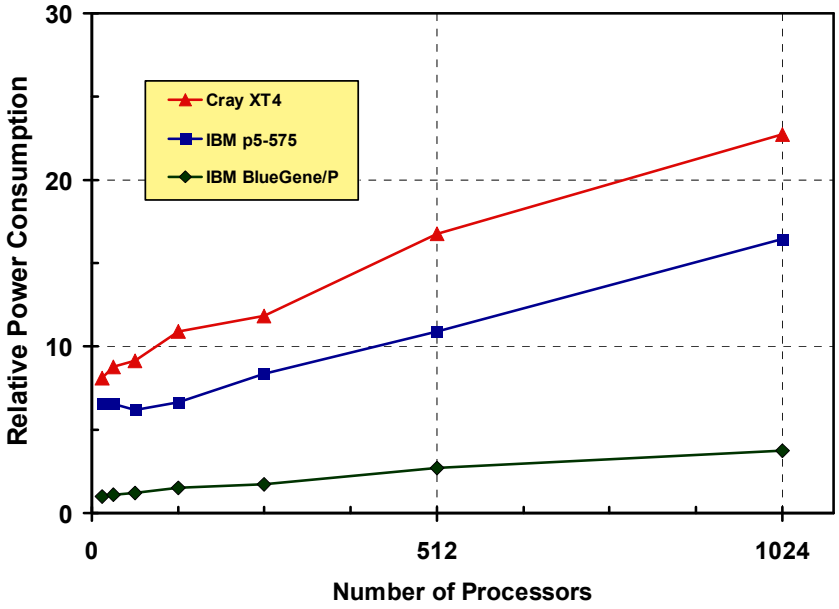


Figure 5. Relative Power Consumption for eigensolve on different HPC platforms (CRYSTAL matrix, n=20480).

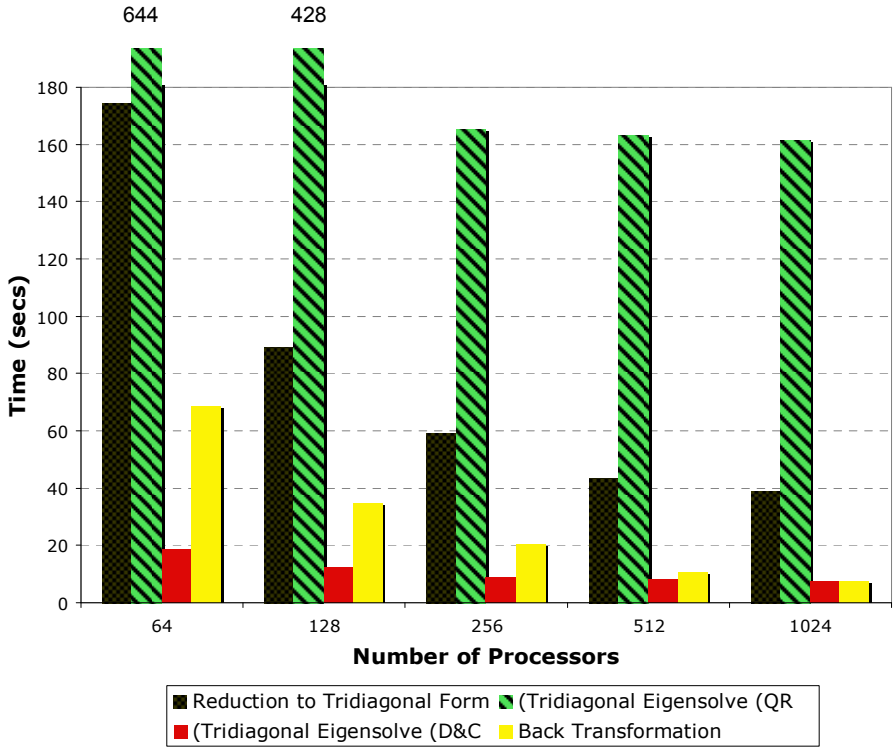


Figure 6. Breakdown of time spent in the constituent stages of dense eigensolvers (CRYSTAL matrix n= 12534, p5-575)

# 5 Conclusions

The latest ScaLAPACK eigensolvers are generally reliable and perform well for the application matrices tested in this paper. Typically, the parallel scaling improves for the larger problem sizes on all the platforms, as the computation to communication ratio increases. In other reports it has been established that both solvers generally perform preferably to the original ScaLAPACK solvers *PDSYEV* and *PDSYEVX* for the matrices under test here [11]. The parallel performance of the pre-release version of the MRRR-based solver *PDSYEVR* obtained from the developers for testing performs comparably to the divide-and conquer based *PDSYEVD* over a range of problem sizes.

On large processor counts where the division of the problem is relatively thin, the performance of *PDSYEVR* appears to degrade somewhat. This problem is addressed in a recent paper [17]. It remains to be seen if the 'holy grail' properties of O(*kn*) operations and memory overheads of O(*n*) are will be achieved for the final release of *PDSYEVR*, hopefully as part of a future release of ScaLAPACK.

Timings from the Cray XT series machines show good parallel scaling can be achieved for larger matrices upto several thousands of processors. The performance results from the new BlueGene architectures sre generally two to three times slower than equivalent parallel runs on the the Cray XT4 for large-scale parallel diagonalizations. This ratio roughly matches that of the respective processor clock speeds (2.8 GHz vs 850 MHz) on the two machines. However, it is now of increasing importance that parallel architectures are power efficient (flops/Watt) in addition to being performance efficient (flops/sec). Figure 5 shows that the power consumption of the Blue Gene/P is around six times lower than the Cray XT4 for a corresponding matrix diagonalization.

If timings for the full symmetric eigenproblem are broken down into the three constituent phases - reduction, tri-diagonal eigensolve and back transformation (Figure 6) - it becomes clear that the tri-diagonal eigensolve may no longer dominate timings. Moreover, the Householder reduction is both relatively slow and scales poorly on large processor counts. This has been recognised by parallel numerical routine developers and new methods are now under investigation to improve the parallel performance of this phase of the calculation [19].

In order to meet the challenges of petascale architectures, where runs may involve tens of thousands of processing cores, it is evident that new parallelization strategies may be required. For example, the PRMAT Hamiltonian matrices represent the wavefunction for a sector of external configuration space defined when calculating the electron-atom scattering problem. A typical problem contains multiple sectors and after some restructuring of the program, the sector Hamiltonian matrix diagonalizations can be calculated concurrently by sub-groups of processors divided up from the global processor population. By taking advantage of this inherent

parallelism in the method it is expected that good parallel scaling of the overall scientific problem could be achieved on many tens of thousands of processors.

## References


[1] *A parallel R-matrix program PRMAT for electron-atom and electron-ion scattering calculations*, A. G. Sunderland, C. J. Noble, V. M. Burke, P. G. Burke, Computer Physics Communications 145 (2002) 311-340.

[2] CRYSTAL: A computational tool for solid state chemistry and physics, www.crystal.unito.it.

[3] K.L. Baluja, P.G. Burke and L.A. Morgan, Comput. Phys. Commun. 27 (1982) 299.

*[4]* *The numerical computation of the characteristic values of a real symmetric matrix*, Wallace J. Givens, Technical Report ORNL-1574, Oak Ridge National Laboratory, Oak Ridge, TN, USA, (1954).

*[5]* *The calculation of specified eigenvectors by inverse iteration*, contribution II/18, volume II of Handbook of Automatic Computation, pagesSpringer-Velag, New Yotk, Heidelberg, Berlin, (1971).

[6] *The QR transformation, parts I and II*, Computer J., 4:265-271, 332-345, (1961-62).

[7] *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).

[8] *A New $O(n^2)$ Algorithm for the Symmetric Tridiagonal Eigenvalue/Eigenvector Problem*, I.S.Dhillon, PhD Thesis, University of California, Berkeley, California, (1997).

[9] *Scalapack Home Page*, www.netlib.org/scalapack/scalapack_home.html

[10] *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.

[11] *Performance of a New Parallel Eigensolver PDSYEVR on HPCx*, A.G.Sunderland, HPCx Technical Report (2006), http://www.hpcx.ac.uk/research/hpc/technical_reports/HPCxTR0608.pdf.

[12] *HECToR - The UK National Supercomputing Service*, www.hector.ac.uk.

[13] *The HPCx Supercomputing Facility*, www.hpcx.ac.uk.

[14] *STFC, The Science and Technology Facilities Council*, www.stfc.ac.uk.

[15] *The IBM BlueGene/L*, www.research.ibm.com/bluegene/.

[16] *The Swiss National Supercomputing Centre*, www-users.cscs.ch/xt4.

[17] *Lapack Working Note 194: A Refined Representation Tree for MRRR*, Christof Voemel, www.netlib.org/lapack/lawnspdf/lawn194.pdf.

[18] *The Green500 List*, www.green500.org.

[19] *Parallel algorithms for reduction of a symmetric matrix to tridiagonal form on a shared memory multiprocessor*, Dogan Kaya, Applied Mathematics and Computation, v169.2, pp1045-1062, 2005