
Parallel Eigensolver Performance on the HPCx System

Andrew Sunderland, Elena Breitmoser

*Terascaling Applications Group
CCLRC Daresbury Laboratory
EPCC, University of Edinburgh*



1. Brief Introduction to HPCx
 - Hardware description.
 - The Terascale Applications Group.
2. Introduction to Eigensolver Routines
 - Numerical Library Eigensolver Routines
 - Parallel Performance Issues
- Brief Summary of Application Codes
- Performance Results
 - Range Of Matrix Sizes
 - Breakdown Timings
- Terascaling Eigensolver-based Applications on HPCx
 - Eigensolver optimisations for HPCx
- Conclusions

Phase 1 (Dec. 2002): 3 TFlop/s Rmax Linpack

- 40 Regatta-H SMP compute systems (1.28 TB memory)
 - 32 X 1.3GHz processors, 32 GB memory; 4 equal partitions (8-way LPAR nodes)
- 2 Regatta-H I/O systems
 - 16 X 1.3GHz processors (Regatta-HPC), 4 GPFS LPARS
 - 2 HSM/backup LPARS, 18TB EXP500 fibre-channel global filesystem
- Switch Interconnect
 - Existing SP Switch2 with "Colony" PCI adapters in all LPARs (20+ usec latency, 350 MBytes/sec bandwidth)
 - Each compute node has two connections into switch fabric (double plane)
 - 160 X 8-way compute nodes in total

Phase 2 (~May 2004): 6 TFlop/s Rmax Linpack

- >40 Regatta-H+ compute systems
 - 32 X 1.7GHz processors, 32 GB memory, full SMP mode (no LPAR)
- 3 Regatta-H I/O systems (Double the capabilities of Phase 1)
- HPS "Federation" switch fabric
 - bandwidth quadrupled, ~5-10 microsecond latency, Connect to GX bus directly

Phase 3 (2006): 12 TFlop/s Rmax Linpack

1. Performance Attributes of Key Applications

- Trouble-shooting with Vampir, gprof

*Terascale Applications
Team within HPCx*

2. Scalability of Numerical Algorithms

- e.g. Parallel eigensolvers

3. Optimisation of Communication Collectives (EPCC)

- MPI_ALLTOALLV

4. Migration from replicated to distributed data

- DL_POLY-3

5. Scientific drivers amenable to Capability Computing

- New Algorithms e.g. Hierarchical MPI /Open MP solutions.

A Computational Bottleneck in Many Scientific Codes

- Ubiquitous in parallel quantum chemistry codes.
- Intensive Communications required.
- Memory Hungry.
- Time consuming - high operation counts.

Standard Eigenvalue Problem:

$$\mathbf{Ax} = \lambda \mathbf{x}$$

.. where **A** is a dense real matrix and λ is an eigenvalue corresponding to eigenvector **x**.

The General Eigenvalue Problem:

$$\mathbf{Ax} = \lambda \mathbf{Bx}$$

.. where **A** and **B** are dense real matrices and λ is an eigenvalue corresponding to eigenvector **x**.

The solution to the real dense symmetric Eigensolver problem usually takes place via three main steps

1. Reduction of the matrix to tri-diagonal form, typically using the Householder Reduction. $O(n^3)$
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, up to $O(n^3)$
 - QR algorithm, up to $O(n^3)$
 - Divide & Conquer method (D&C), up to $O(n^3)$
 - Multiple Relatively Robust Representations (MR3 algorithm). $O(nk)$
3. Back substitution to find Eigenvectors for the full problem. $O(n^3)$

Other Methods

- Jacobi Method $O(n^3)$
- Symmetric Subspace Decomposition Algorithm

- **Scalapack**
 - drivers for solving standard and generalized dense symmetric or dense Hermitian Eigenproblems.
 - PDSYEV (QR Method) (Scalapack 1.5)
 - PDSYEVX (Bisection & Inverse Iteration) (Scalapack 1.5)
 - PDSYEVD (D&C Method) (Scalapack (1.7))
- **Peigs**
 - General symmetric and standard symmetric eigenproblems
 - PDSPEV (uses parallel subspace iteration method)
- **BFG**
 - Block Jacobi Method on full dense symmetric matrix (Hermitian also)
- **Plapack**
 - QR method
 - MRRR 'Multiple Relatively Robust Representations'

Drivers for solving standard and generalized dense symmetric or dense Hermitian Eigenproblems

- Real Symmetric Eigenproblem Driver Routines
 - PDSYEV & PDSYEV D calculate all eigenvalues & eigenvectors.
 - PDSYEVX (Xpert driver)
- Routines for solving non-symmetric Eigenproblem
- ScaLAPACK requires the following libraries:
 - Blacs, PBlas, MPI, Lapack, Blas (Pessl)
 - Block-cyclic distribution of data across processors
- Precompiled versions of the ScaLAPACK library are available for several platforms. (www.netlib.org)
- The memory requirements for the solvers are generally of $O(n^2)$.
- Matrices with tightly clustered Eigenvalues require repeated, i.e. costly, re-orthogonalisations.

Pei GS provides both standard and generalized dense, real and tri-diagonal symmetric Eigensystem drivers

- PDSPEV (real symmetric - all eigenvalues & eigenvectors)
- PDSPEVX (selection)
- Both use Householder, Bisection & Inverse Iteration, Back Substitution
 - Peigs3.0 includes Dhillon & Partlett's optimisations for the inverse iteration stage
- Memory requirements $O(n^2)$

Column based distribution of data

- Mapping vector describes column distribution across processors

Libraries Used

- MPI (or Global Array Toolkit), Lapack, Blas (Essl)

Peigs is included in the NWCHEM package

- Available from the Pacific Northwest National Laboratory

Previous investigations reported that it is generally slower than ScaLAPACK, but scales better on parallel machines.

BFG solves the standard dense symmetric, real and Hermitian Eigenproblems

- Iterative method based on a Block Jacobi Eigensolver
- Written in Fortran 90
- Uses MPI, BLAS (Essl)
- Interfaces for column-based and block-cyclic data distributions

BFG performance issues

- slower in serial than Householder based methods but shows excellent scaling.
- ideal for multiple calls to an Eigensolver, where previous solutions are a fair approximation of subsequent solutions.

Availability

- Upon request from Ian Bush, Daresbury Laboratory, i.j.bush@dl.ac.uk

- PLAPACK solves for dense symmetric standard Eigenproblems.
 - QR algorithm
 - MRRR (MR³) algorithm
 - Written in Fortran and C
 - Uses Abstract Programming Interface
 - Data distributed by columns

Performance Issues:

- MR³ generally requires only $O(n^2)$ operations and $O(n)$ workspace.
 - Smaller memory overheads allow larger problem sizes

Availability

- Plapack can be downloaded from the web
- MR³ routine (beta version) available upon request.
 - No drivers available
- Initial reports suggest the library offers **good load balancing** and **excellent scaling** for larger cases.

- Electronic structure and related properties of periodic systems
- All electron, local Gaussian basis set, DFT and Hartree-Fock

→ • $H^R = P^R \cdot I^R$
 $I^R \leftarrow$ sum of independent integrals

• $H_k \leftarrow Q_k^T H^R Q_k$

• $H_k \psi_k = \epsilon_k \psi_k$

- Solve $H_k \Rightarrow \{\epsilon_k, \psi_k\}$

• $P^R \leftarrow |\psi_k|^2$

• Repeat until converged

I^R - functionally parallel

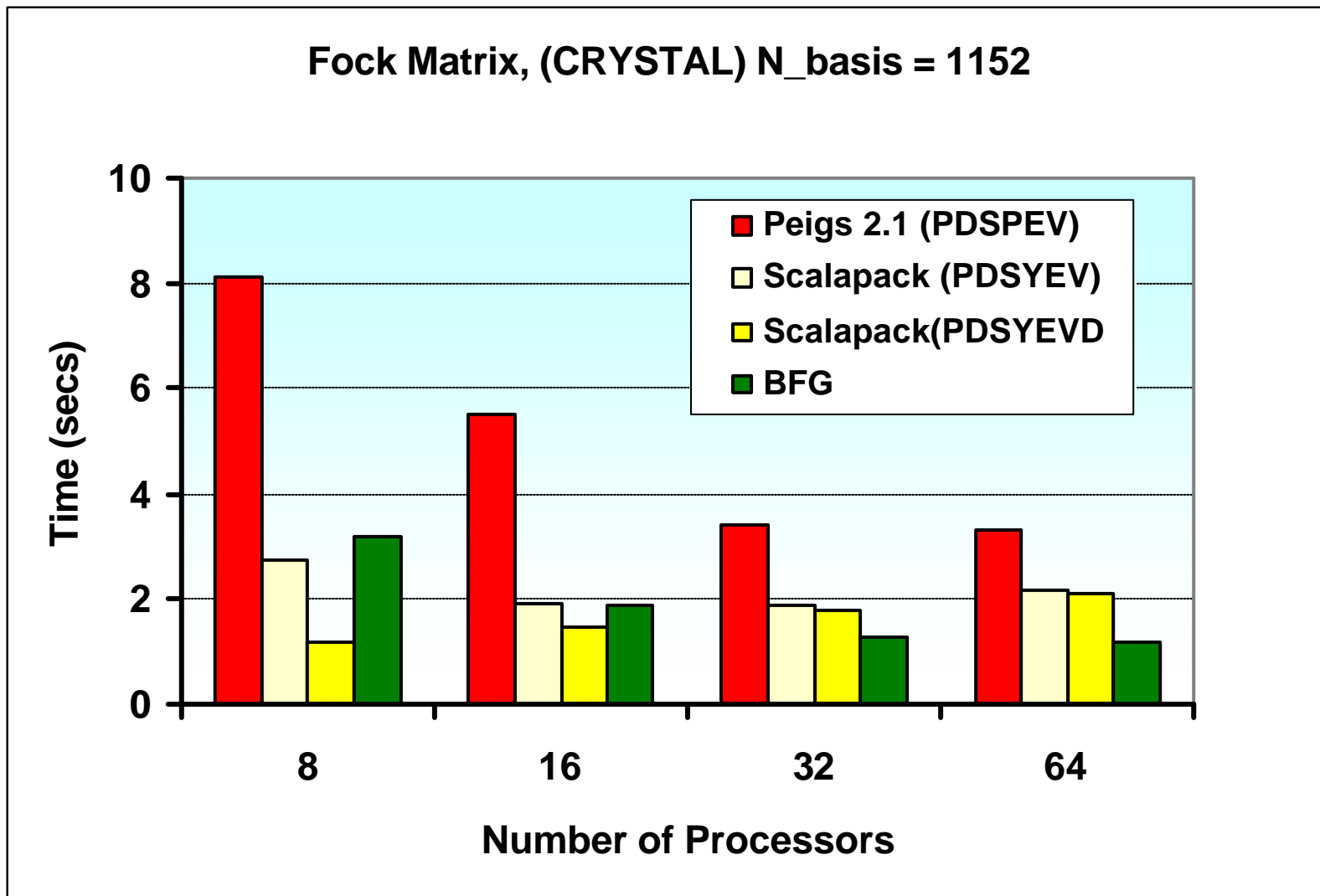
Global sum H^R

F.T. and distributed
matrix multiply

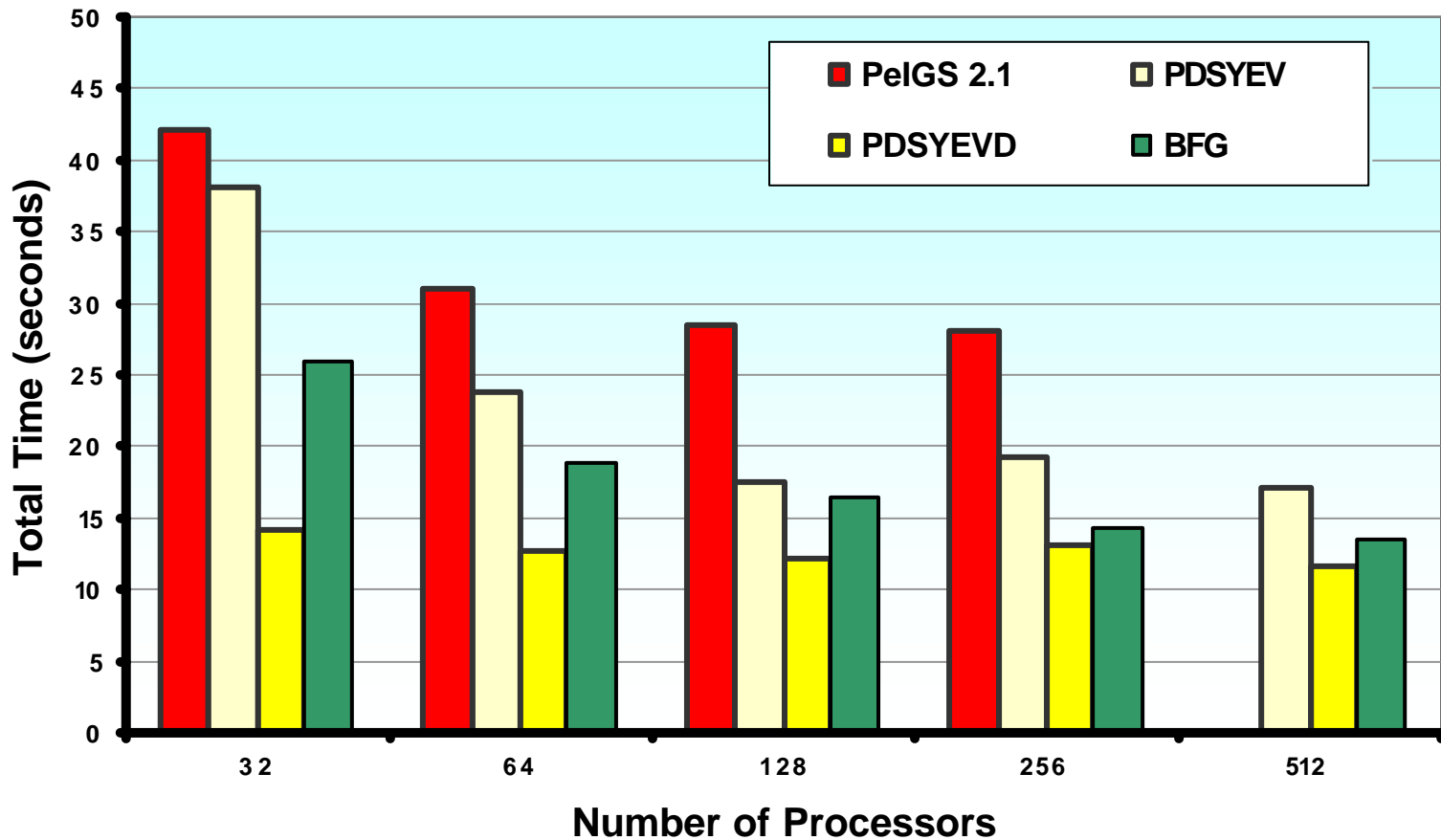
Distributed diag

BFG Jacobi

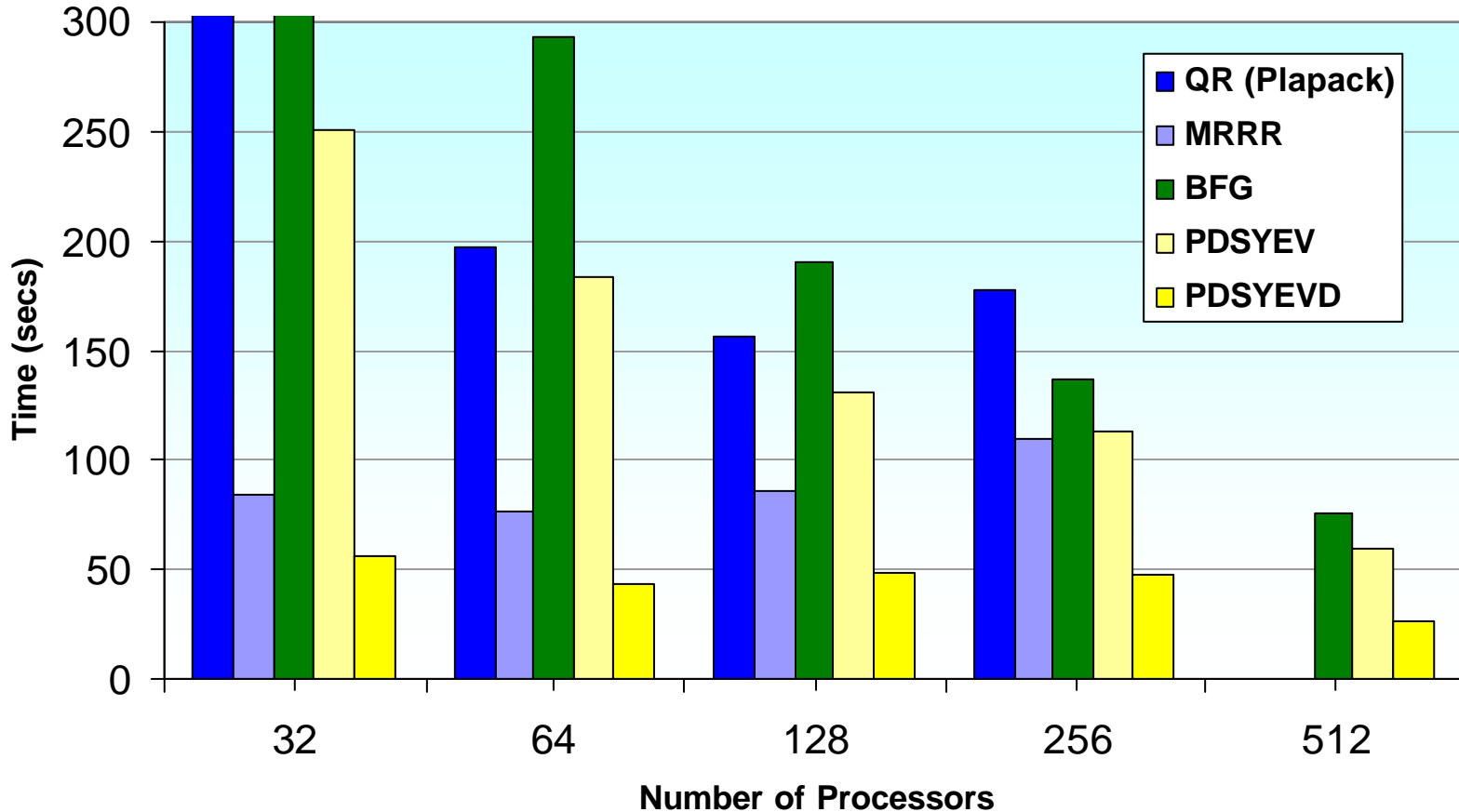
Gather & condense



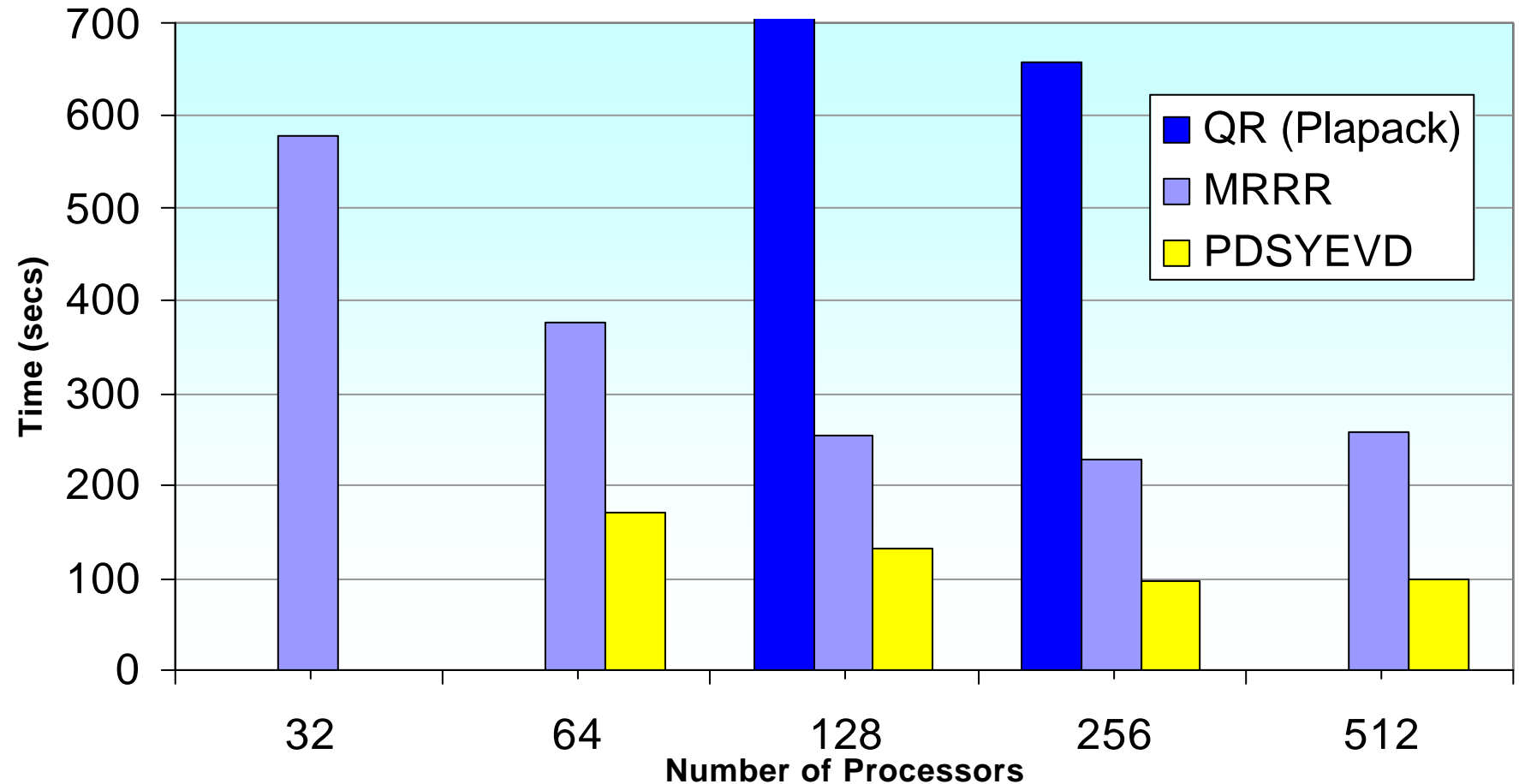
Fock Matrix, (CRYSTAL), N_basis = 3888



Fock Matrix (Crystal), N_basis=7194



Fock Matrix, (Crystal) N_basis=12354



- PDSYEVD performs better than PDSYEV for all problem sizes investigated
 - PDSYEVD fast, but needs large problems to scale well
- Peigs performs fairly well for small matrices
- BFG slower for small processor counts, but scales very well to high processor counts
- MR³ getting better for larger matrices, but still slower than PDSYEVD

- Electronic structure and related properties of periodic systems
- All electron, local Gaussian basis set, DFT and Hartree-Fock

→ • $H^R = P^R \cdot I^R$
 $I^R \leftarrow$ sum of independent integrals

• $H_k \leftarrow Q_k^T H^R Q_k$

• $H_k \psi_k = \epsilon_k \psi_k$

- Solve $H_k \Rightarrow \{\epsilon_k, \psi_k\}$

• $P^R \leftarrow |\psi_k|^2$

• Repeat until converged

I^R - functionally parallel

Global sum H^R

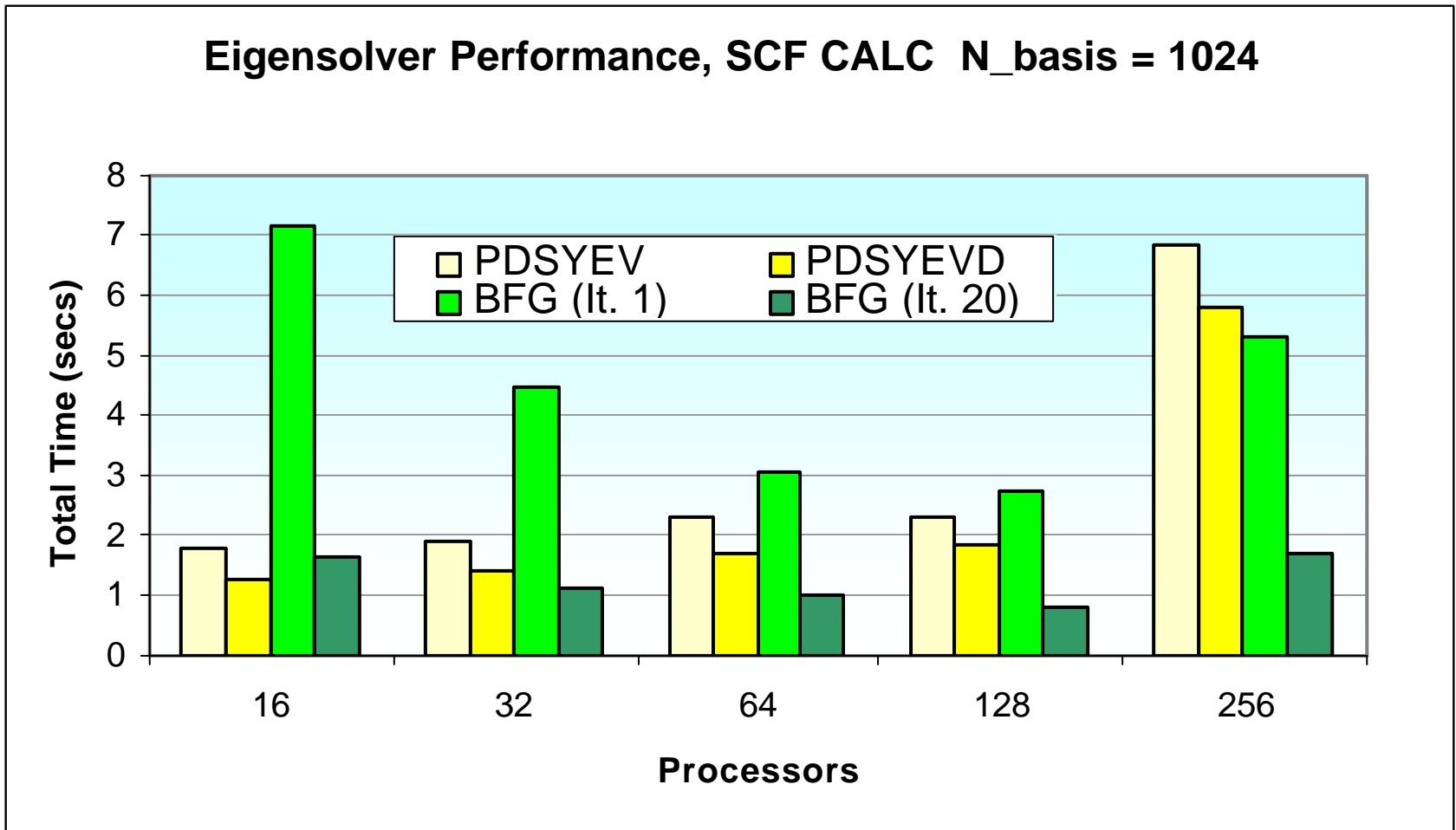
F.T. and distributed
matrix multiply

Distributed diag

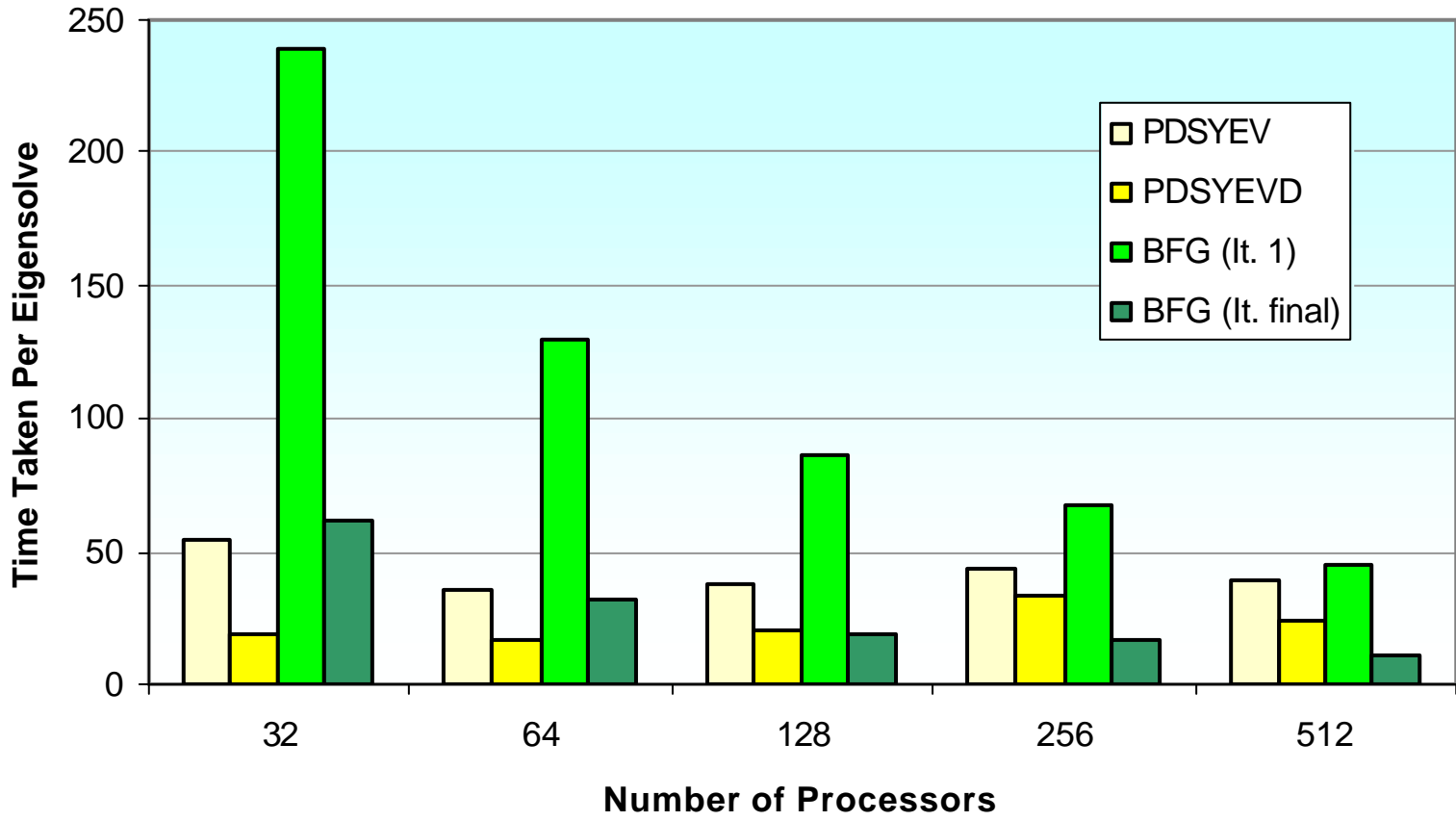
BFG Jacobi

Gather & condense

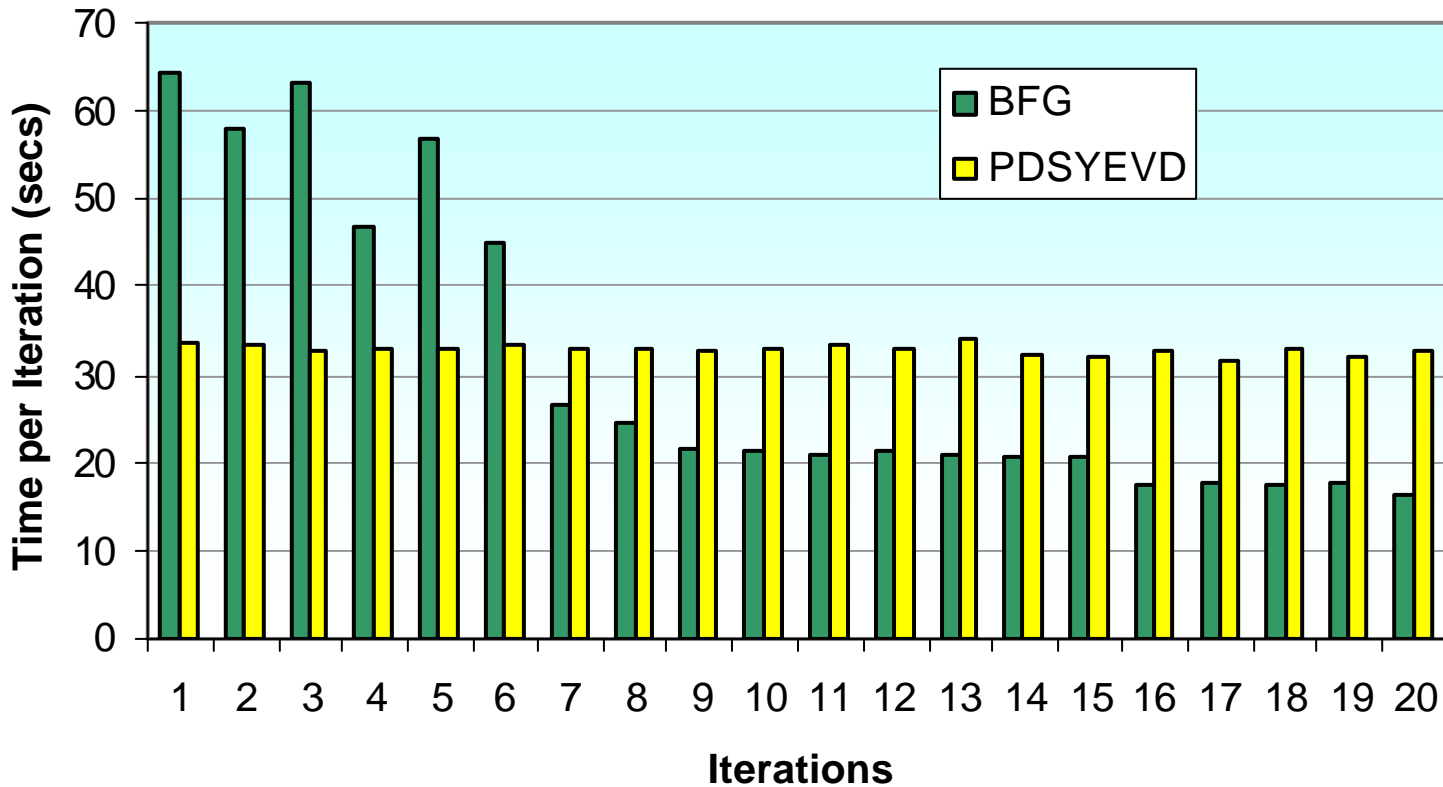
Eigensolver Performance, SCF CALC N_basis = 1024

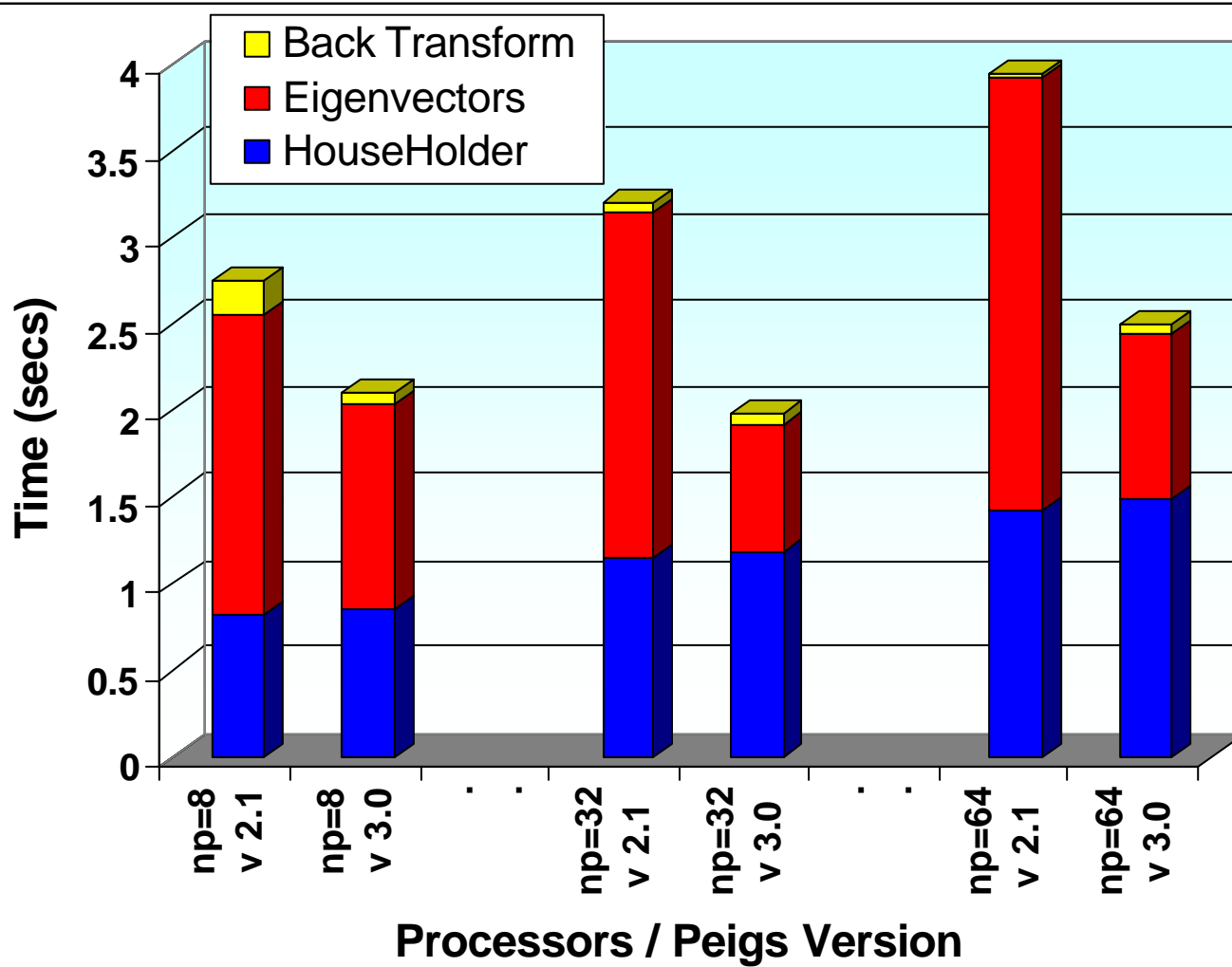


Eigensolver Performance, SCF CALC , N_basis = 4096

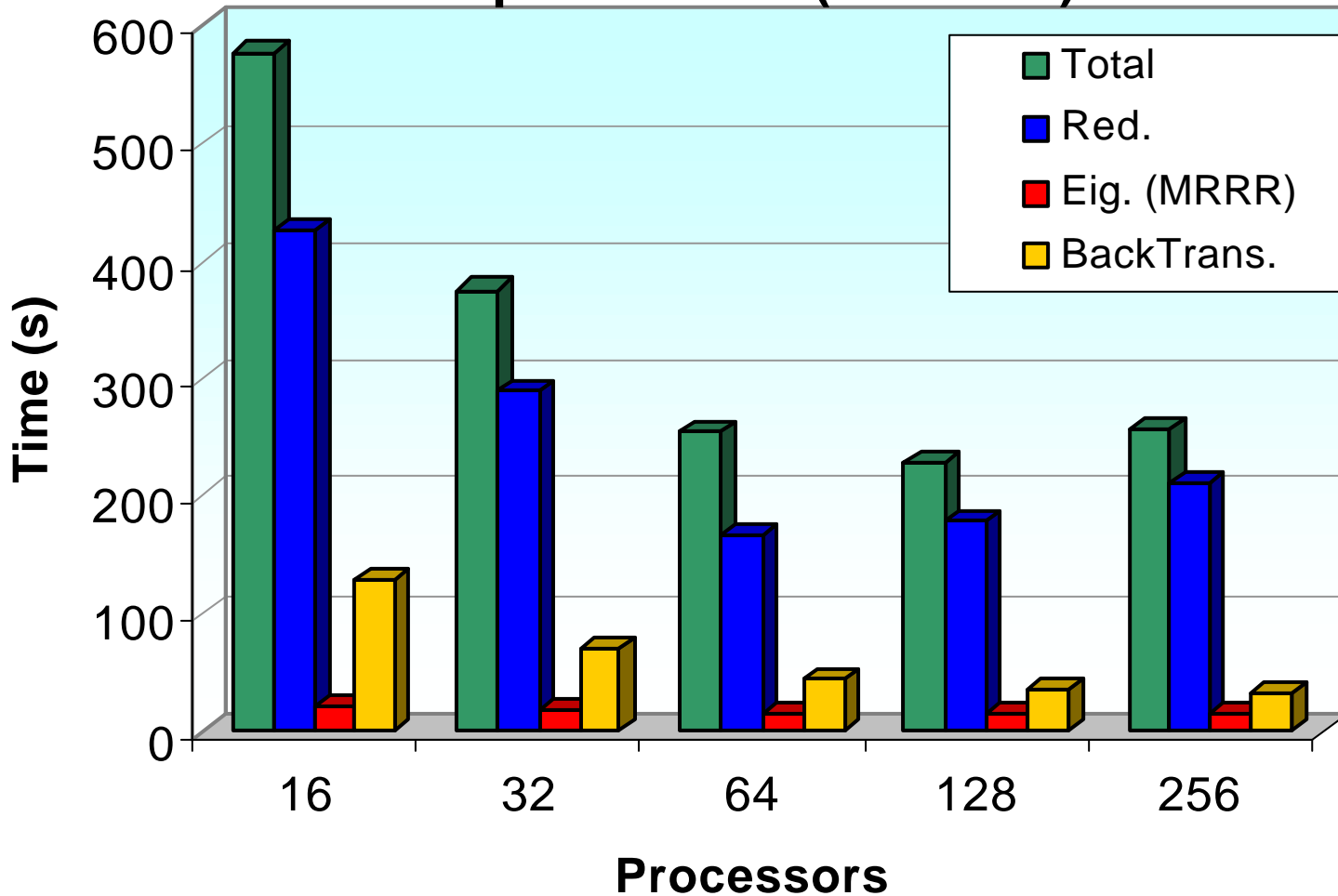


**Eigensolver Performance, SCF CALC 4096 basis functions
(256 processors)**

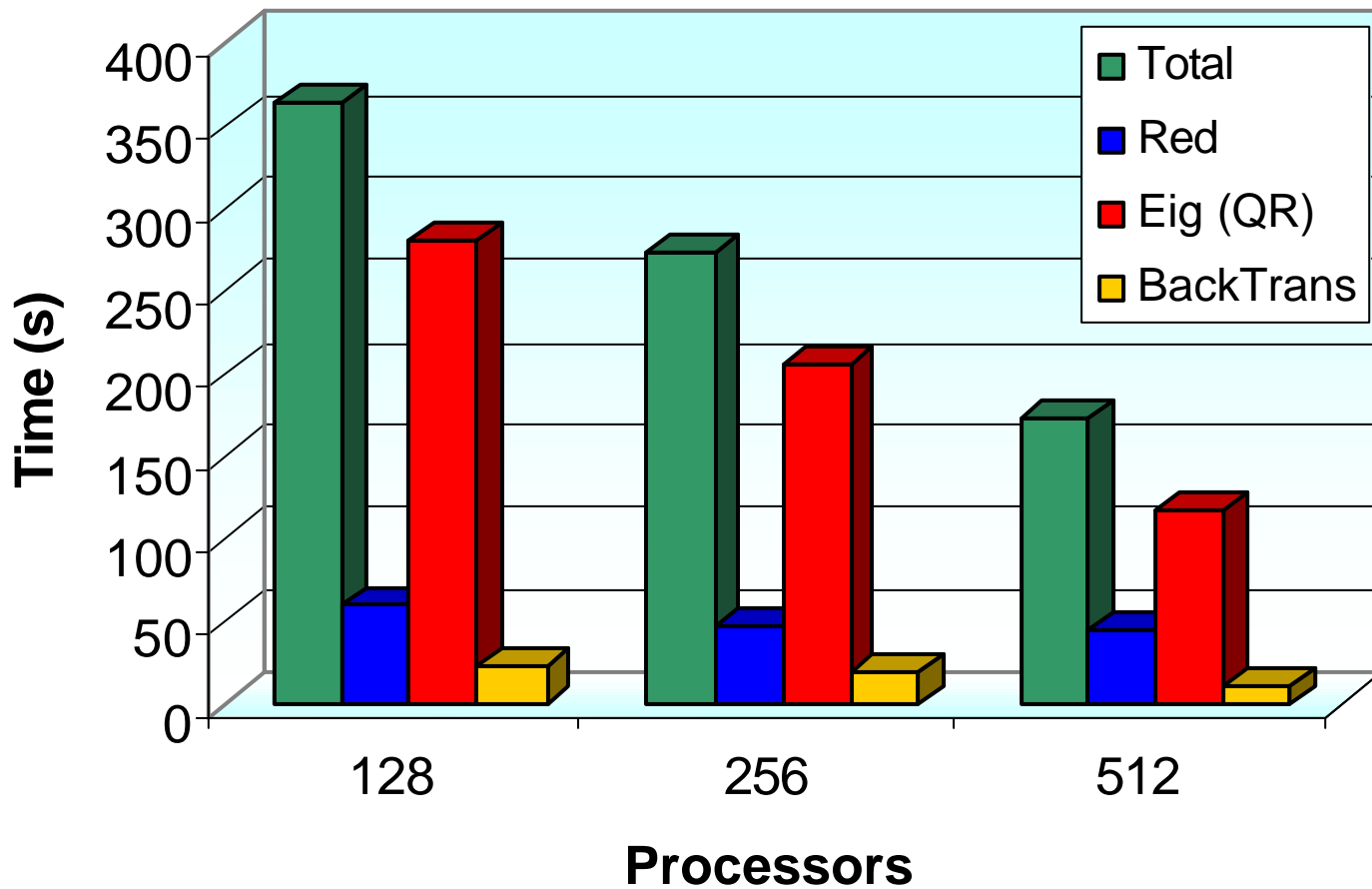




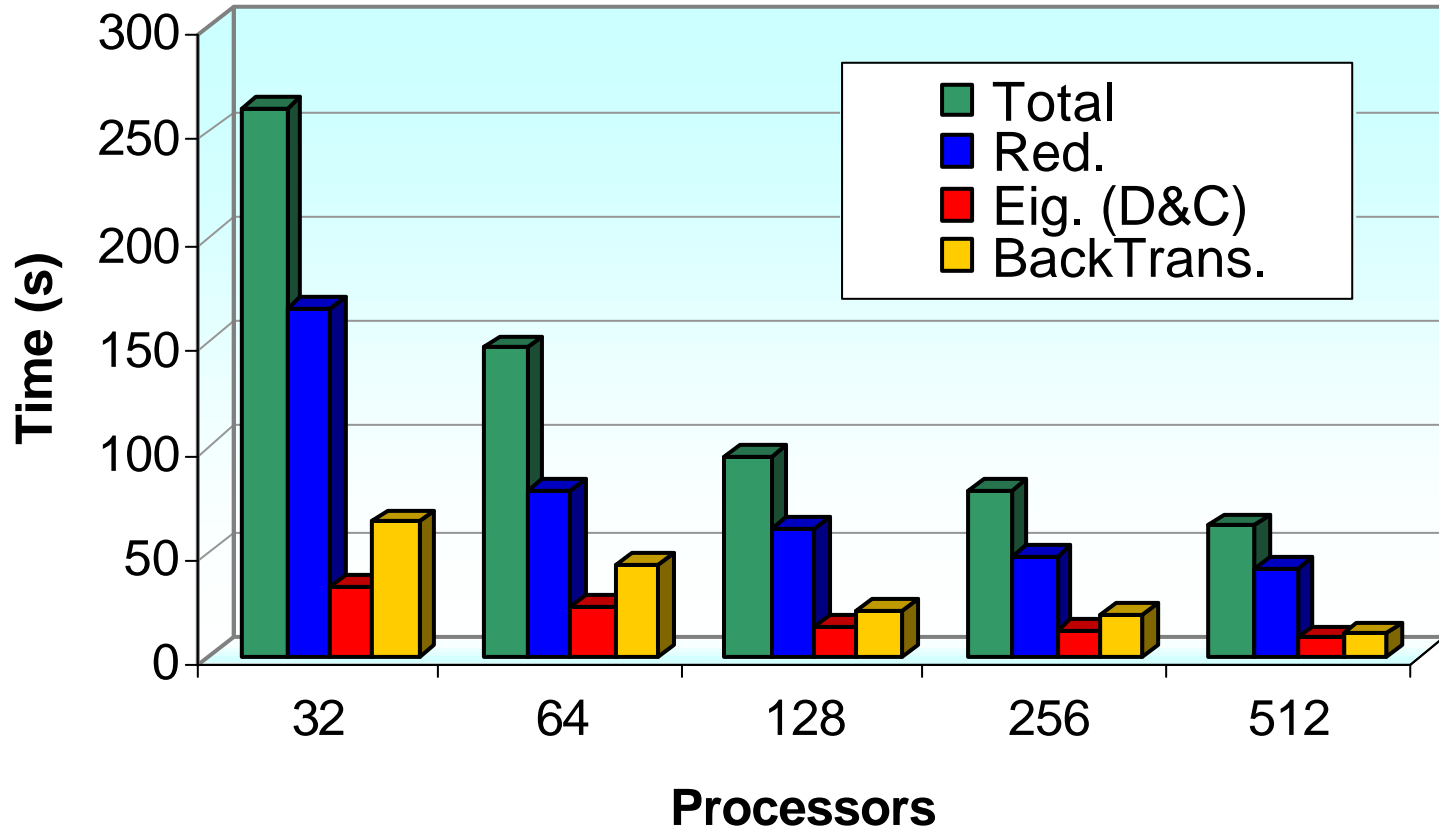
Plapack MRRR (N=12354)



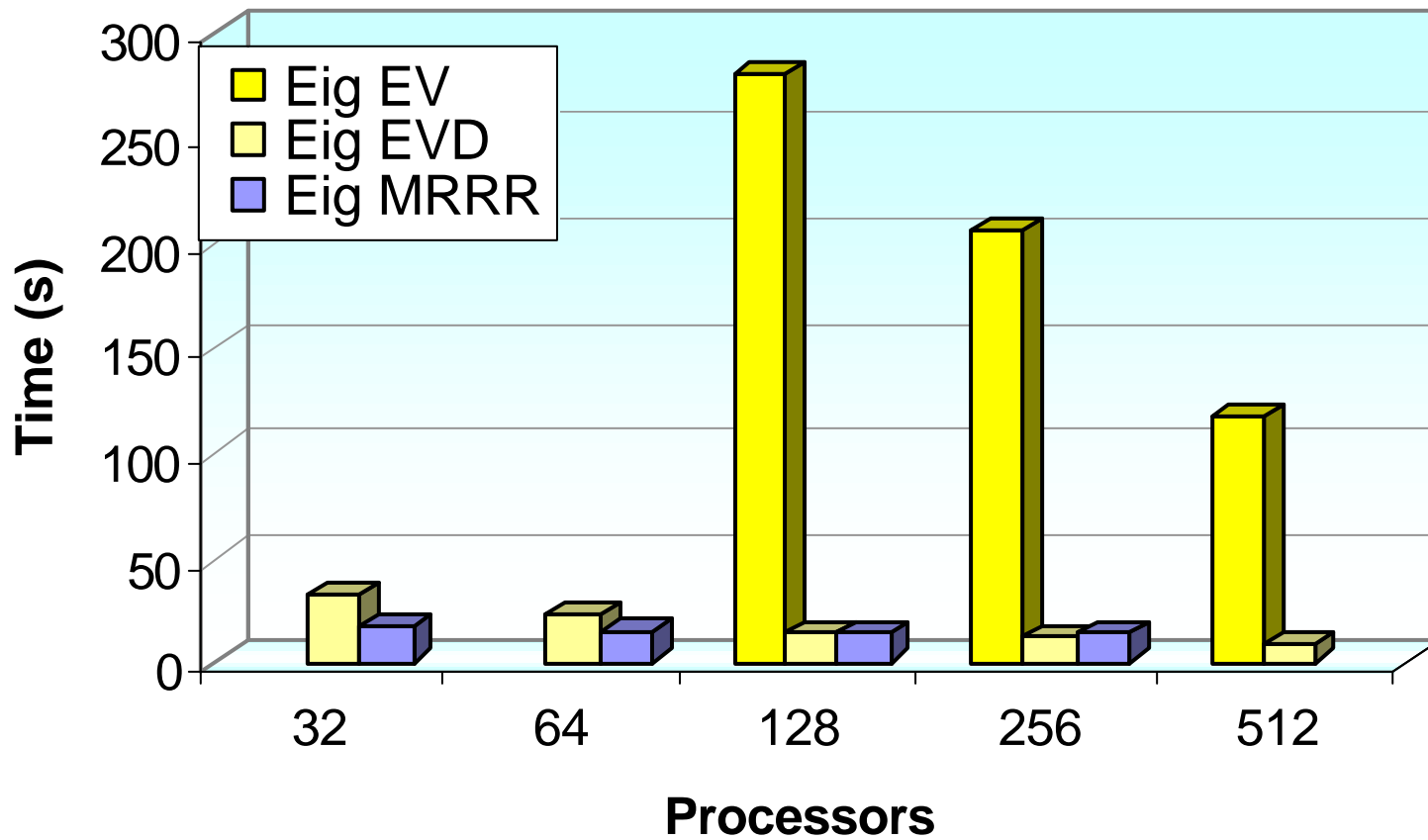
Scalapack PDSYEV (N=12354)



Scalapack PDSYEVD (N=12354)

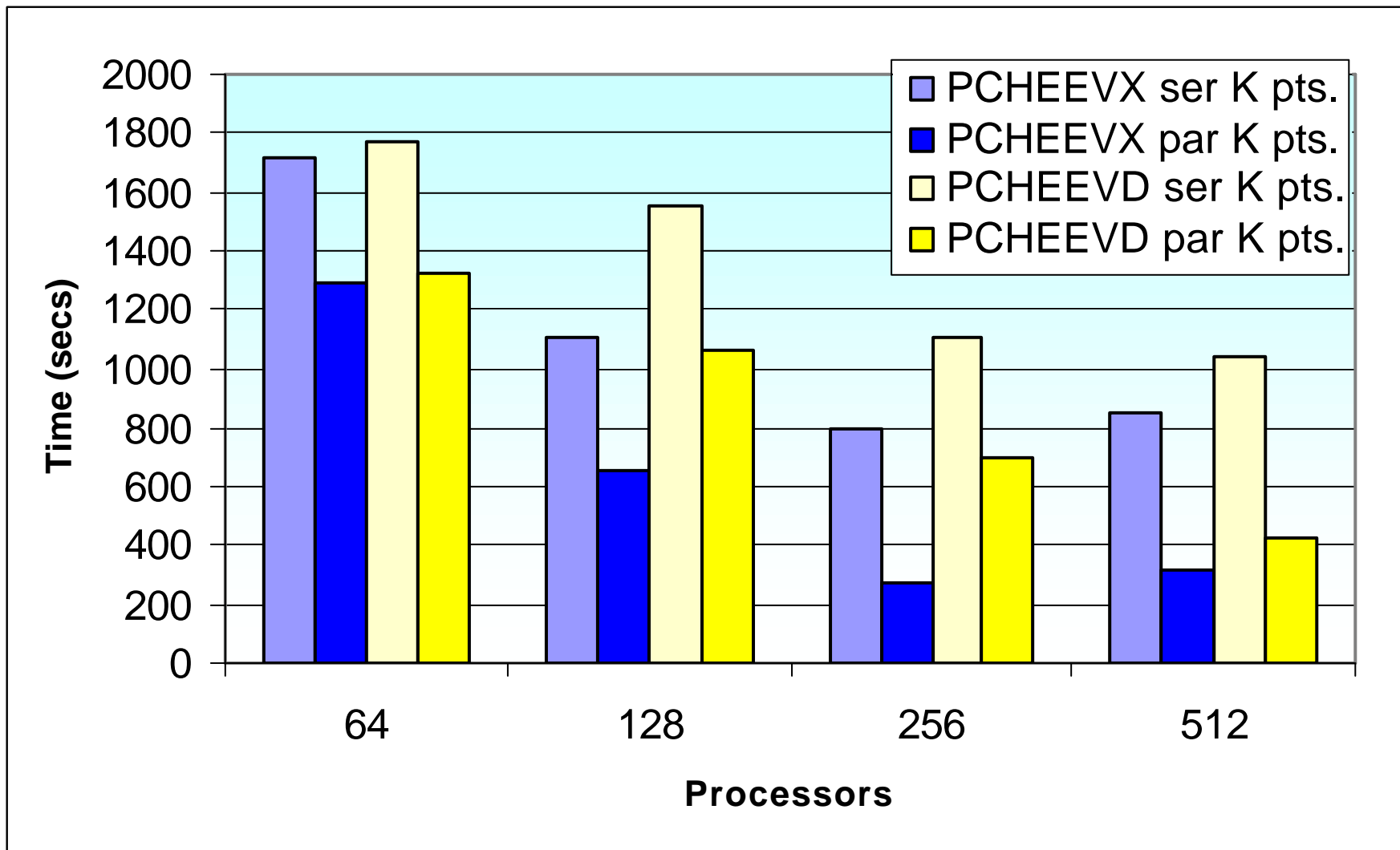


Eigenvalue & Eigenvector Calculation (N=12354)

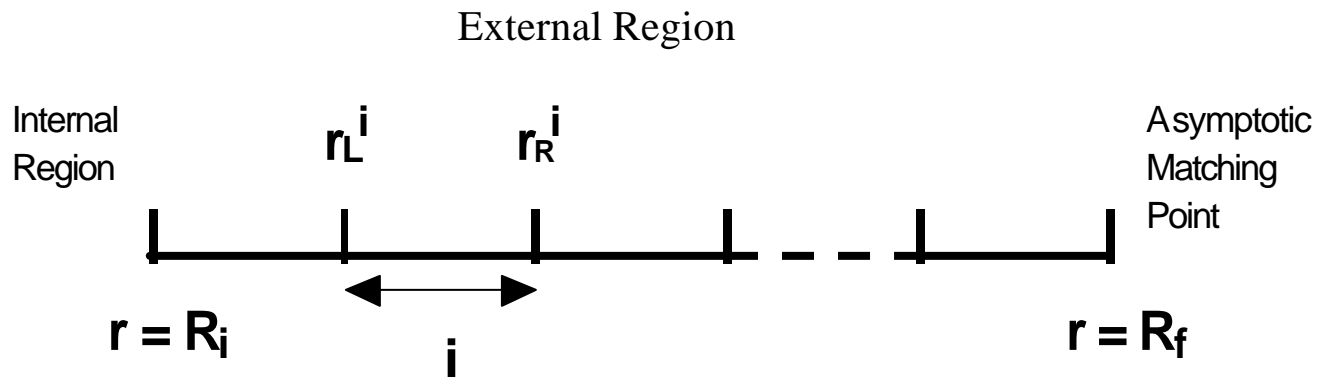


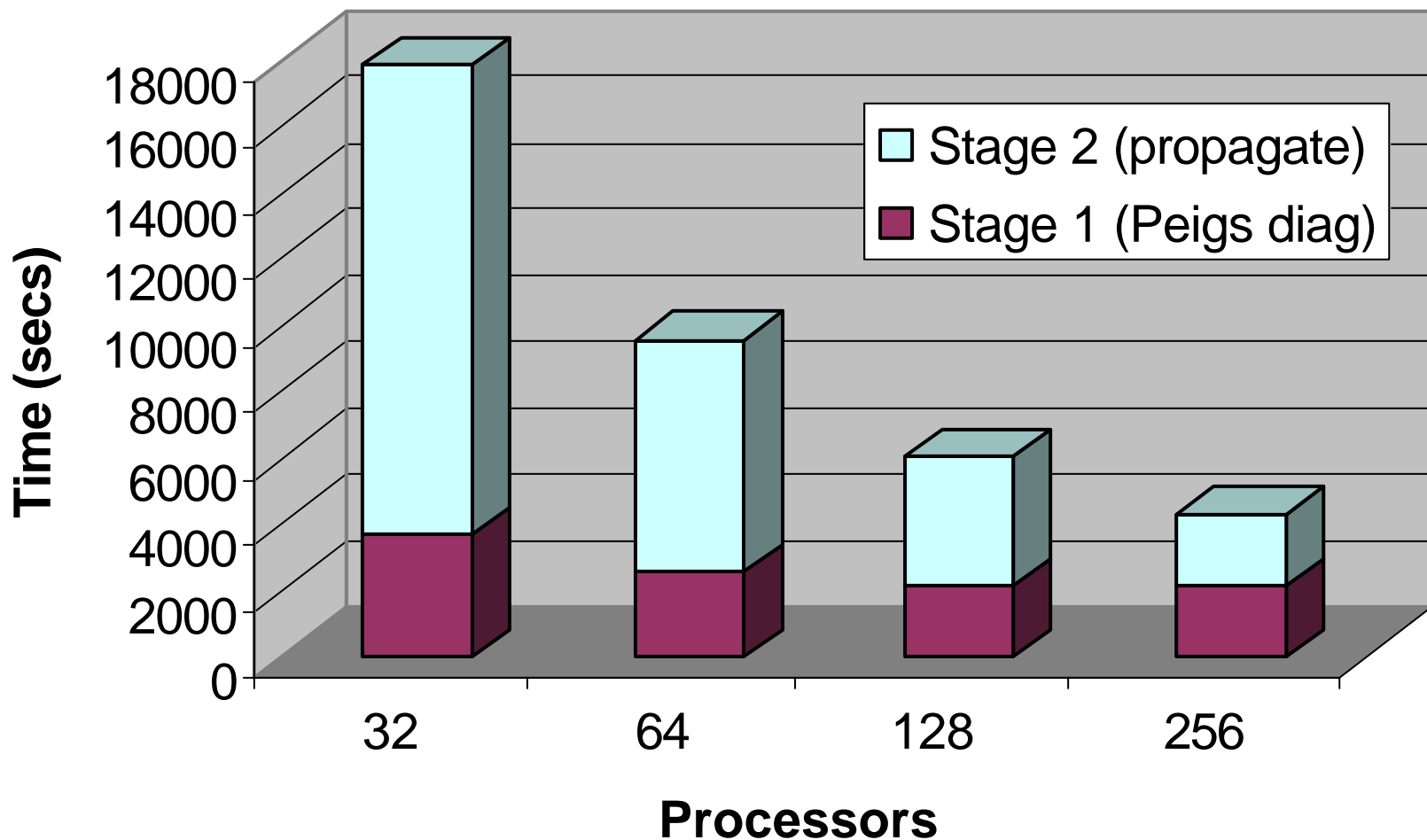
AIMPRO (Ab Initio Modelling PROgram)

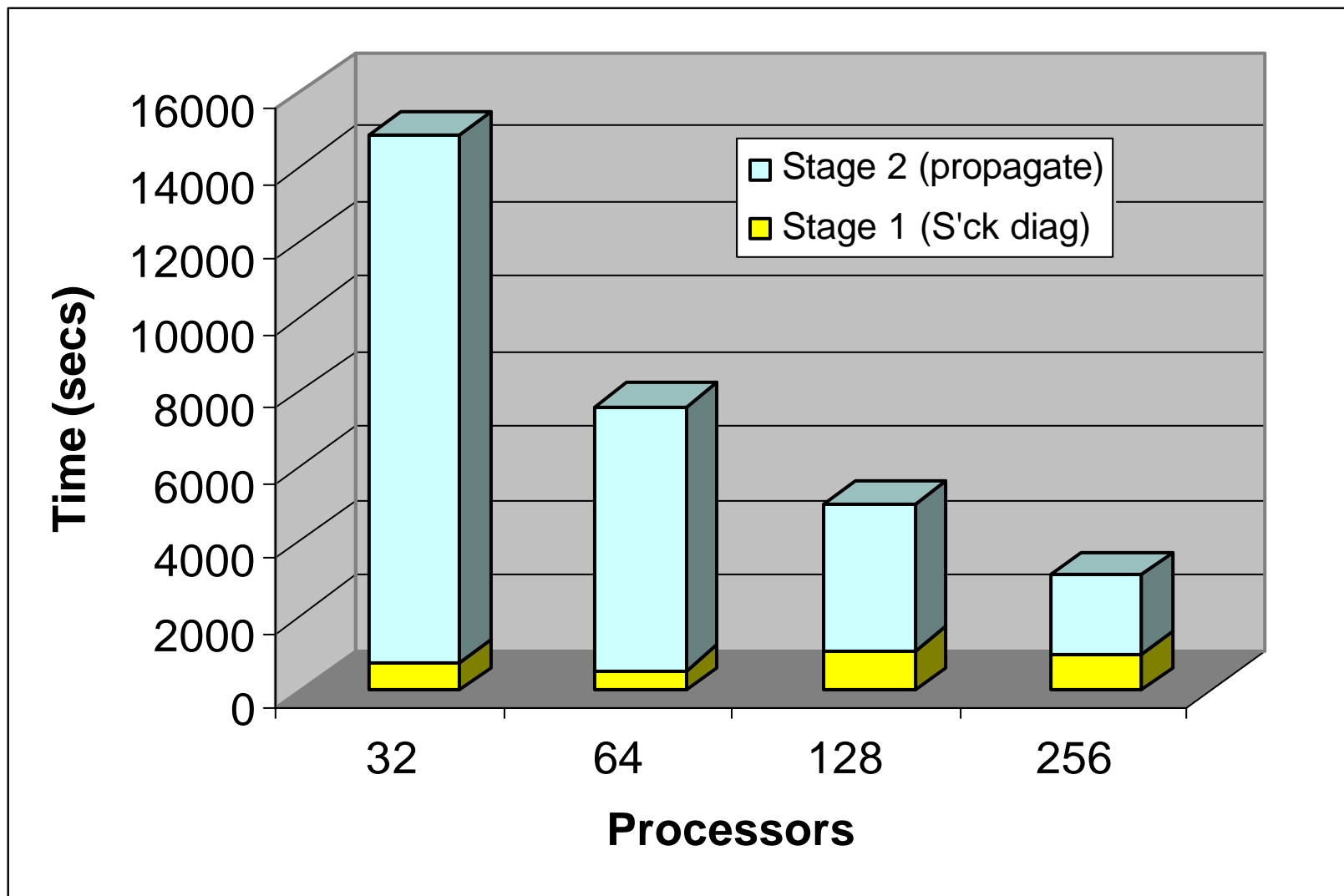
- developed by Patrick Briddon et al. at Newcastle University.
- studies both molecules and 3D periodic systems
- uses a Gaussian basis set to solve the Kohn-Sham equations for the electronic structure.
- Calculation dominated by **Parallel Diagonalizations** & Parallel Matrix Multiplies
- Utilises **K - point parallelism**
- Uses Scalapack complex (PCHEEV(X/D)) and real (PDSYEV(X/D)) symmetric eigensolver driver routines

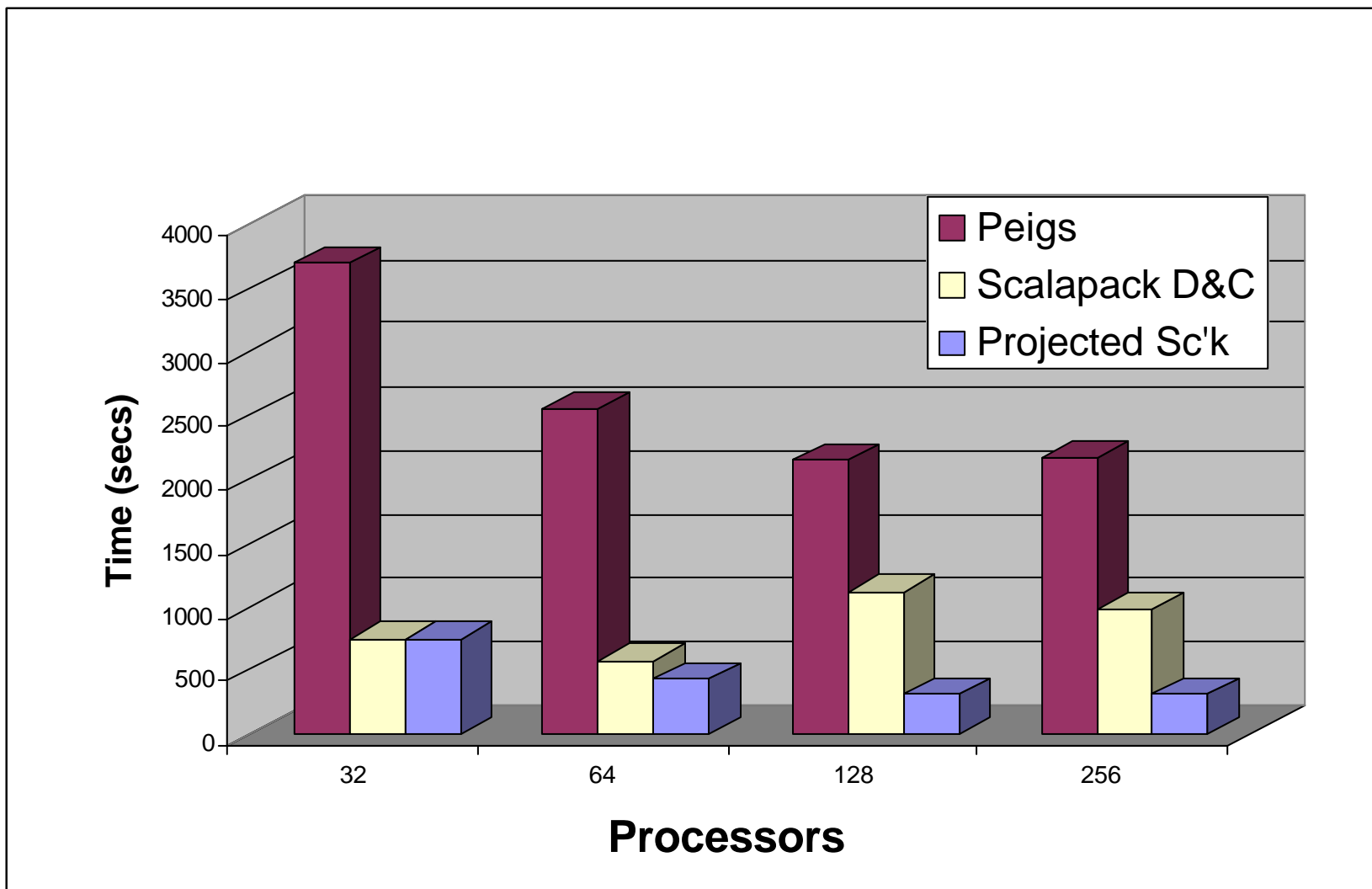


- R-matrix theory provides efficient computational methods for investigating electron-atom and electron-molecule collisions.
- Divides external atomic region into several sectors
- Calculation involves integration of large sets of coupled second-order linear differential equations.
- 2 Stage Calculation - Stage 1 requires Diagonalizations of large Sector Hamiltonian Matrices.









Scalapack Divide & Conquer Routine - best general performer

- Workspace permitting

Peigs slower on low proc. counts but scales fairly well.

BFG for multiple, related diagonalizations

- Slow for serial/low processor counts, but scales well.
- Highly accurate

Plapack MR³

- MR³ performs well, but poor parallel reduction implementation.
- Low workspace overheads, better for large matrices
- Still under development

- Eigenvalue/vector calculation stage parallelised
 - Reduction to tridiagonal form is a computational bottleneck
 - "A Parallel Algorithm for the Reduction to Tridiagonal Form for Eigendecomposition" Siam J. Sci. Comp. Vol. 21 No. 3 pp. 987-1005.
- Other parallelisations can be used to promote scaling:
 - K-point parallelism (AI MPRO)
 - Sector diags. in parallel (PRMAT)
- Initial HPS-based timings appear promising

- " An Overview of Eigensolvers on HPCx", A.G.Sunderland, E. Breitmoser
 - <http://www.hpcx.ac.uk/research/hpc/>
- CLRC Web Description of BFG solver:
 - <http://www.cse.clrc.ac.uk/arc/bfg.shtml>
- Peigs Documentation
 - www.emsl.pnl.gov/docs/nwchem/doc/peigs/docs/peigs3.html
- PLAPACK
 - <http://www.cs.utexas.edu/users/plapack>
- Forthcoming Publication on Eigensolver Performance on HPCx
 - <http://www.hpcx.ac.uk/research/hpc/>

- Elena Breitmoser, EPCC - (MR³)
- Ian Bush, CCLRC Daresbury Laboratory (Crystal, SCF)
- Patrick Briddon, University of Newcastle (Aimpro)
- Cliff Noble, CCLRC Daresbury Laboratory (PRMAT)