JÜLICH FORSCHUNGSZENTRUM

A Comparison of Parallel Dense Symmetric Eigensolvers

on the IBM BlueGene/P System JUGENE

September 28, 2010 | Elin Solberg

Outline

- Introduction
- The eigensolvers tested
- The tests performed
- Results of performance testing
- Todo

Introduction

Definition

Let A be an $n \times n$ matrix. Then a scalar λ is called an eigenvalue and a non-zero (column) vector v the corresponding eigenvector of A if $Av = \lambda v$.

An eigenvector and its corresponding eigenvalue are together referred to as an eigenpair.

Application example

Density Functional Theory (DFT)

- 20-25% of eigenpairs of large dense matrices computed

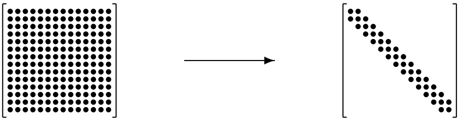
JÜLICH
FORSCHUNGSZENTRUM

BlueGene/P System JUGENE

- 72 racks with 32 nodecards $\times$ 32 compute nodes (total 73728)
  - Compute node: 4-way SMP processor
  - Core type: 32-bit PowerPC 450 core 850 MHz
  - Cores: 294912
- Overall peak performance: 1 Petaflops
- Main memory: 2 Gbytes per node
- Networks:
  - Three-dimensional torus (compute nodes)
  - Global tree / Collective network (compute nodes, I/O nodes)

Three-Phase Algorithm

1. Reduction of the full matrix to tridiagonal form



2. Solution of the tridiagonal eigenproblem
3. Back-transformation of eigenvectors

The Eigensolvers

PDSYEV (ScaLAPACK)

QR algorithm

PDSYEVX (ScaLAPACK)

Bisection and inverse iteration,
optionally computes only a part of the eigenpairs

PDSYEVD (ScaLAPACK)

Divide and conquer

PDSYEVR (Christof Vömel, Zurich)

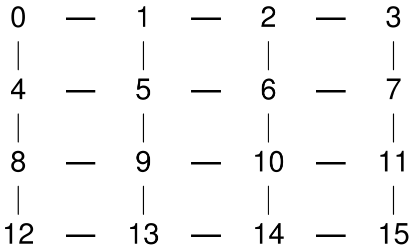
Multiple Relatively Robust Representations (MRRR),
optionally computes only a part of the eigenpairs

PDSYEVX and the Parameter ORFAC

- Symmetric matrix $\Rightarrow$ orthogonal eigenvectors (in theory)
- Clustered eigenvalues $\Rightarrow$ non-orth. eigenvectors with PDSYEVX
- Reorthogonalized if the eigenvalues are 'close'
- Reorthogonalization done on one processor per cluster $\Rightarrow$ poor scaling
- Done for eigenvalues closer to each other than ORFAC $\cdot \|A\|$
  - ORFAC = 0 $\Rightarrow$ no reorthogonalization
  - ORFAC = 10^{-3} (default) $\Rightarrow$ bad performance
  - ORFAC = 10^{-4} used in this project (orthogonality ok)

Matrix Storage Using ScaLAPACK

- Block-cyclic distribution



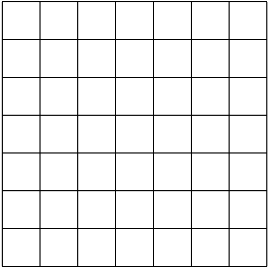
0	1	2	3	0	1	2
4	5	6	7	4	5	6
8	9	10	11	8	9	10
12	13	14	15	12	13	14
0	1	2	3	0	1	2
4	5	6	7	4	5	6
8	9	10	11	8	9	10

Small blocks ⇒ much communication
Large blocks ⇒ bad load balancing (sometimes)

PDSYEV and Block Sizes

Old grid:

```
0  — 1  — 2  — 3
|    |    |    |
4  — 5  — 6  — 7
|    |    |    |
8  — 9  — 10 — 11
|    |    |    |
12 — 13 — 14 — 15
```



New grid:

0 - 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15

JÜLICH
FORSCHUNGSZENTRUM

PDSYEV and Block Sizes

Old grid:

0 — 1 — 2 — 3

4 — 5 — 6 — 7

8 — 9 — 10 — 11

12 — 13 — 14 — 15

0	1	2	3	4	5	6
0	1	2	3	4	5	6
0	1	2	3	4	5	6
0	1	2	3	4	5	6
0	1	2	3	4	5	6
0	1	2	3	4	5	6
0	1	2	3	4	5	6

New grid:

0 - 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15

PDSYEV and Block Sizes

Old grid:

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

0	1	2	3	4	5	6
0	1	2	3	4	5	6
0	1	2	3	4	5	6
0	1	2	3	4	5	6
0	1	2	3	4	5	6
0	1	2	3	4	5	6
0	1	2	3	4	5	6

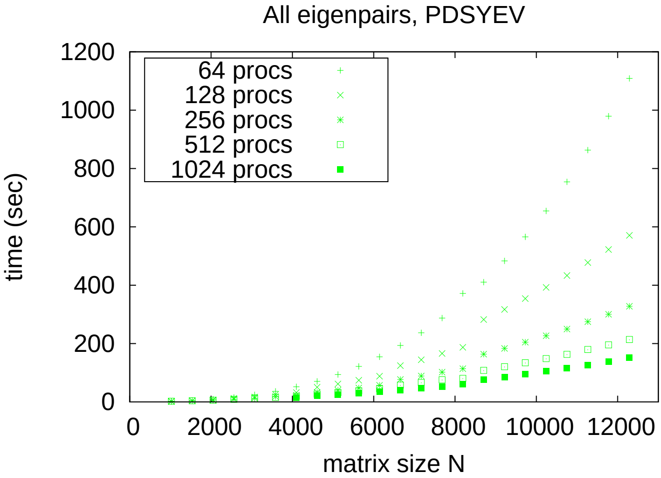
New grid:

0 - 1 - 2 - 3 - 4 - 5 - 6 - **7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15**

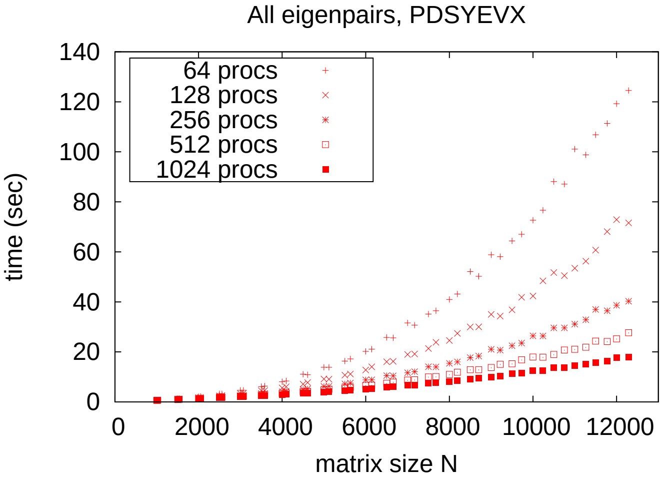
Tests Performed

- Matrix sizes:
  - 1000, 12000, +500
  - 1024, 12288, +512
- Matrix spectra:
  1. All eigenvalues randomly generated in the intervals [0,1], [-10,20], [-100,200], [-100,2000] and [-100,20000]
  2. $\lambda_1, \ldots, \lambda_{N-1} \approx \varepsilon, \ \lambda_N = 1$
- Processor numbers: 64, 1024, *2
- Part of eigenpairs:
  - All eigenpairs with all solvers
  - 10% of the eigenpairs with EVX and EVR

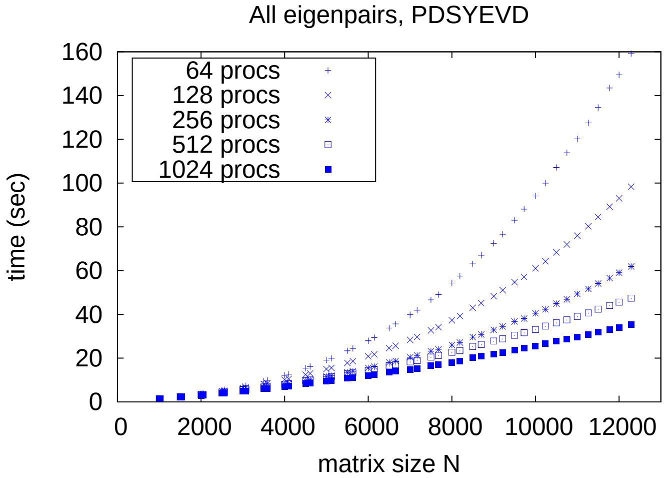
Runtimes, Eigenvalues In [0,1]



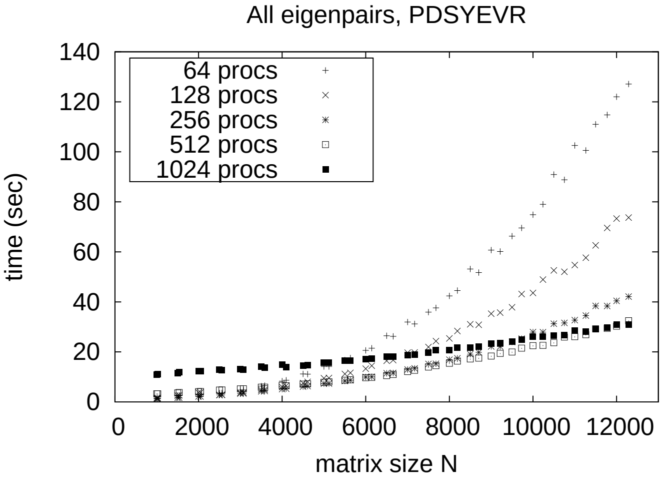
Runtimes, Eigenvalues In [0,1]



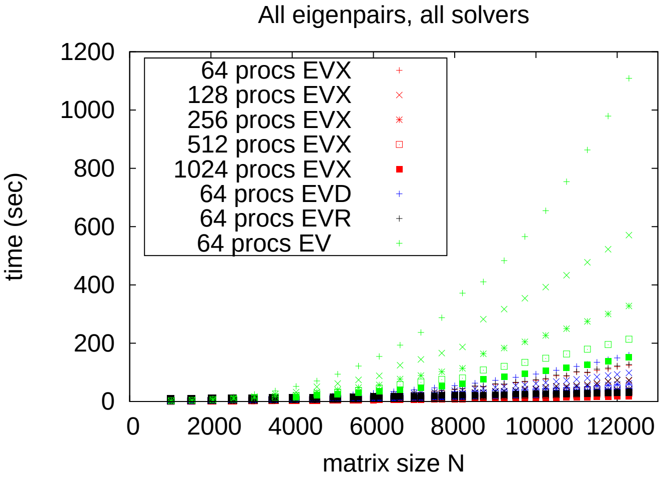
Runtimes, Eigenvalues In [0,1]



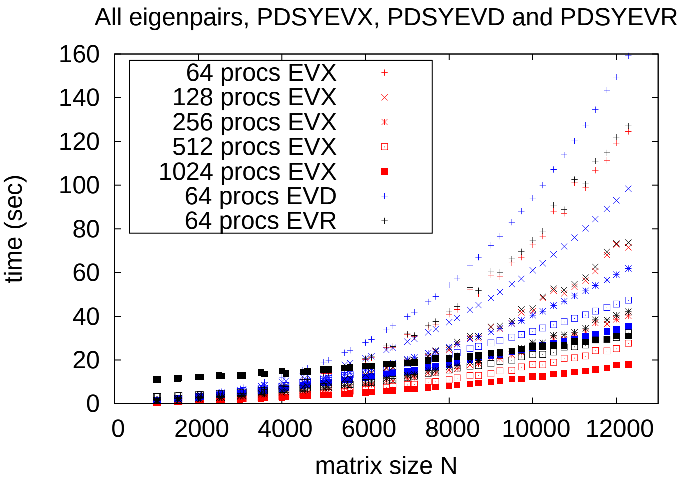
Runtimes, Eigenvalues In [0,1]



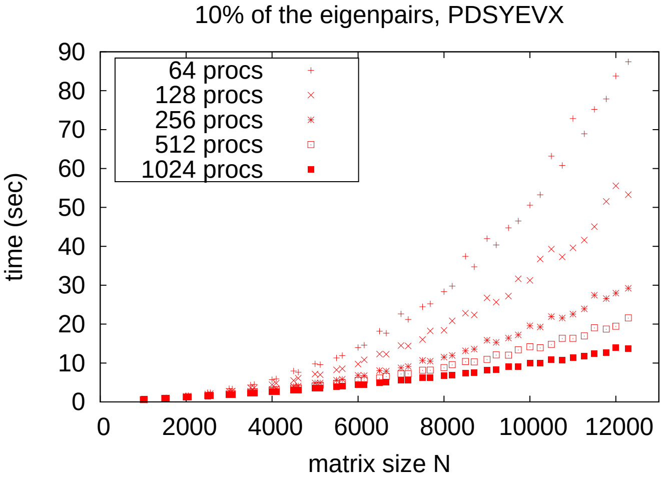
Runtimes, Eigenvalues In [0,1]



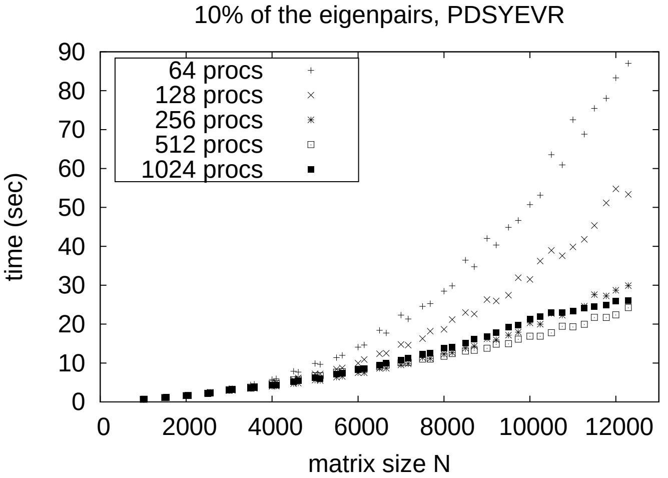
Runtimes, Eigenvalues In [0,1]



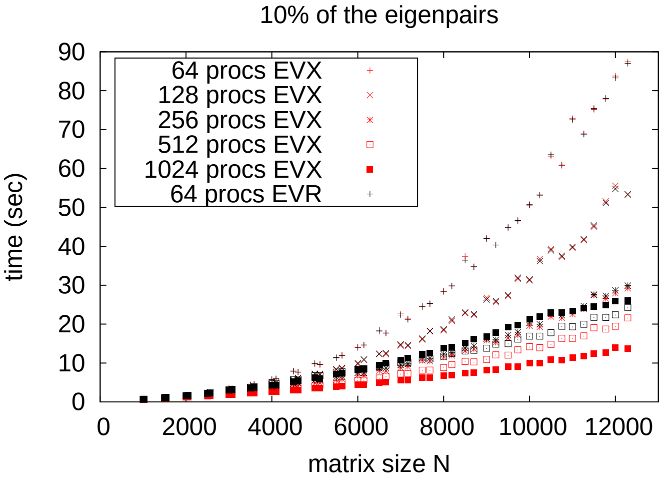
Runtimes, Eigenvalues In [0,1]



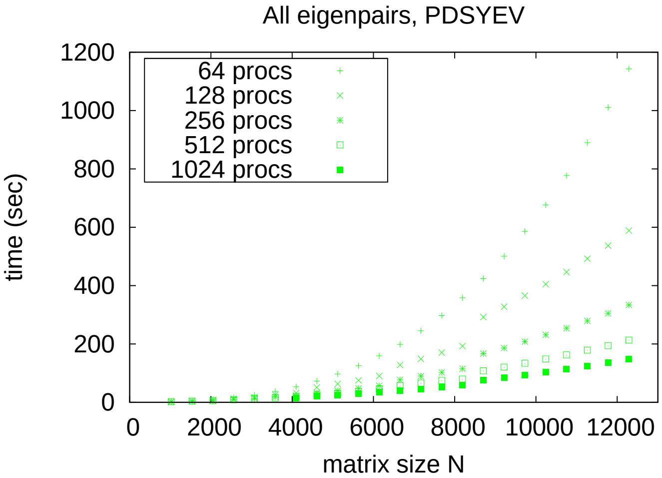
Runtimes, Eigenvalues In [0,1]



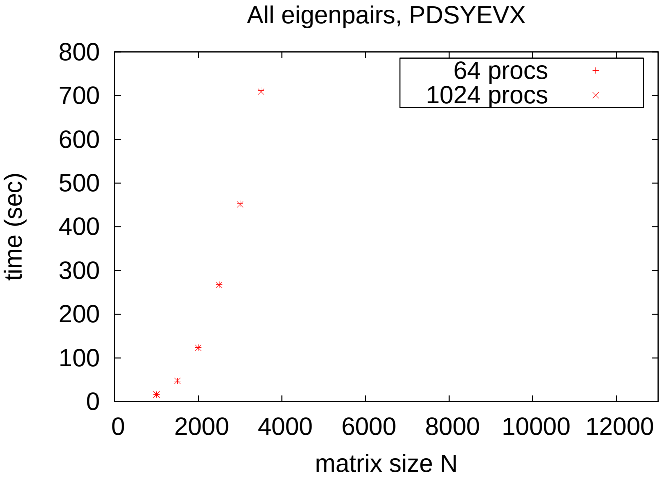
Runtimes, Eigenvalues In [0,1]



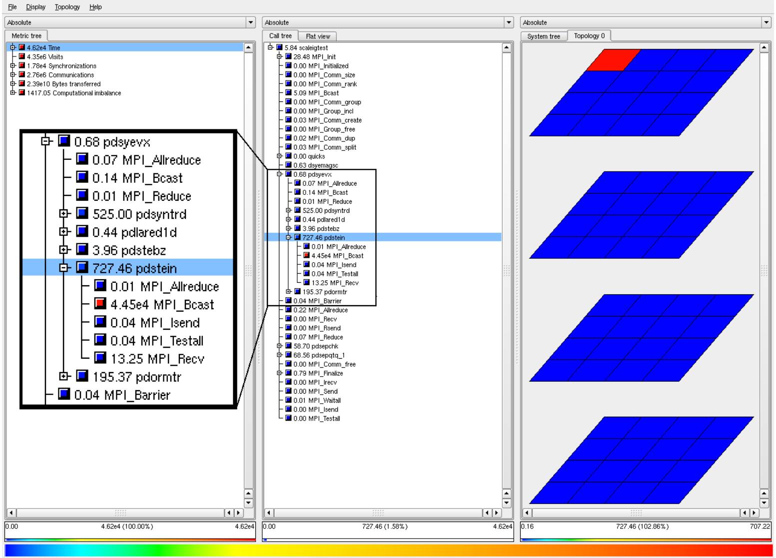
Runtimes, Eigenvalues Clustered Around ε



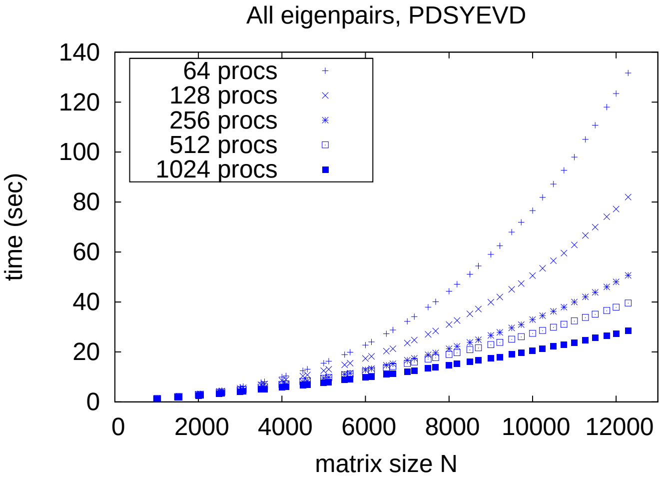
Runtimes, Eigenvalues Clustered Around ε



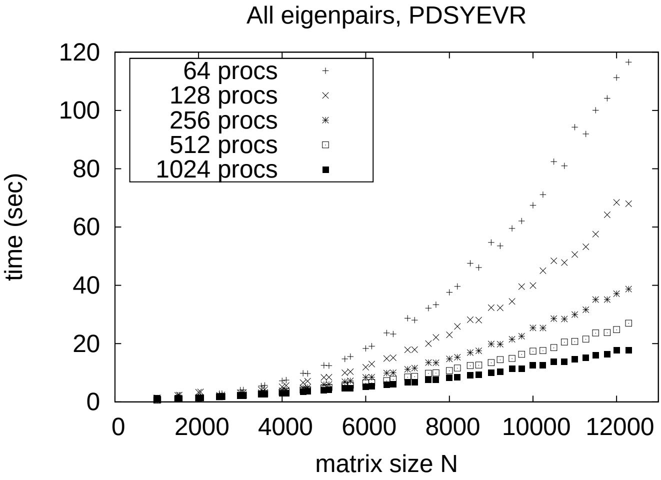
PDSYEVX With Clustered Eigenvalues



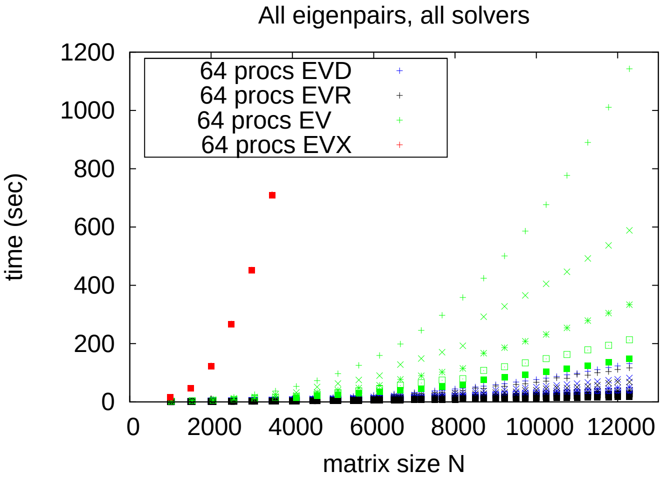
Runtimes, Eigenvalues Clustered Around ε



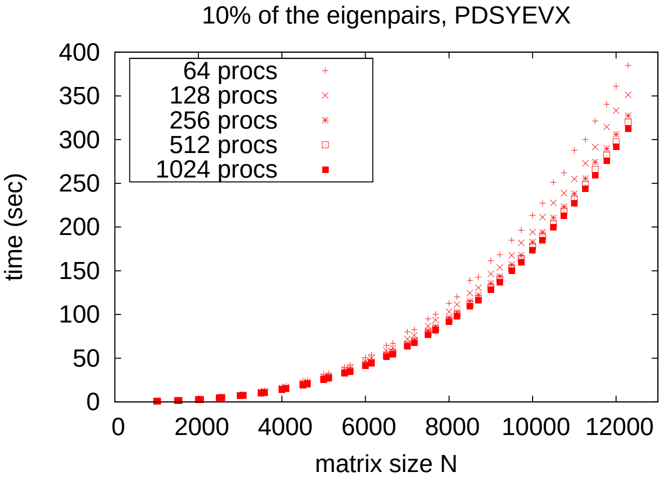
Runtimes, Eigenvalues Clustered Around ε



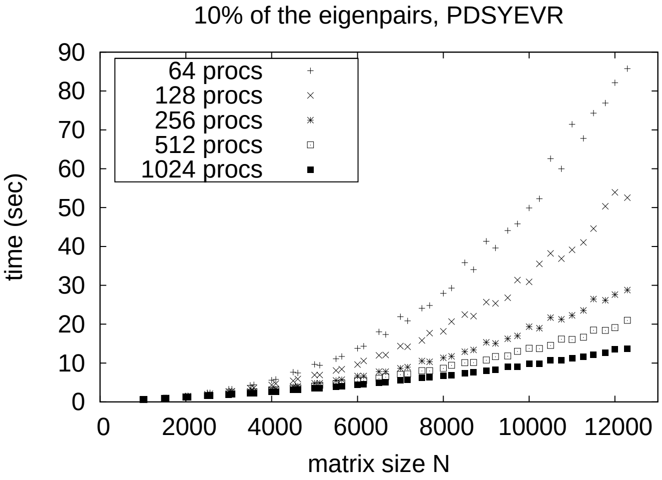
Runtimes, Eigenvalues Clustered Around ε



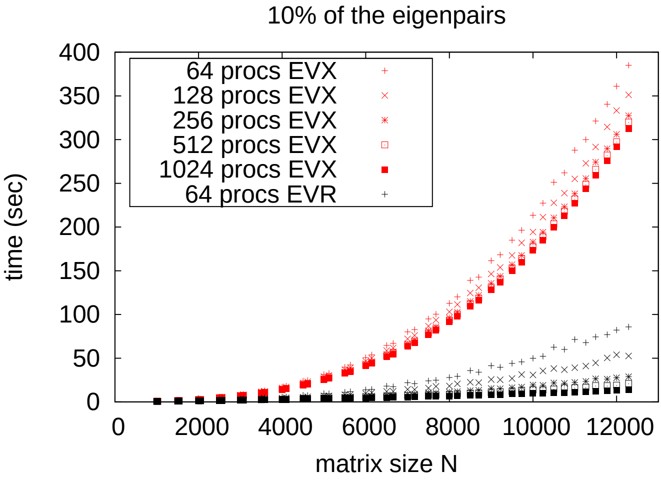
Runtimes, Eigenvalues Clustered Around ε



Runtimes, Eigenvalues Clustered Around ε



Runtimes, Eigenvalues Clustered Around ε



Highlights

- If using PDSYEV, consider block size carefully
- Using PDSYEVX with ORFAC=10^{-4} orthogonality was good enough
- PDSYEVR promising alternative to PDSYEVX for computing a part of the eigenpairs when eigenvalues are clustered

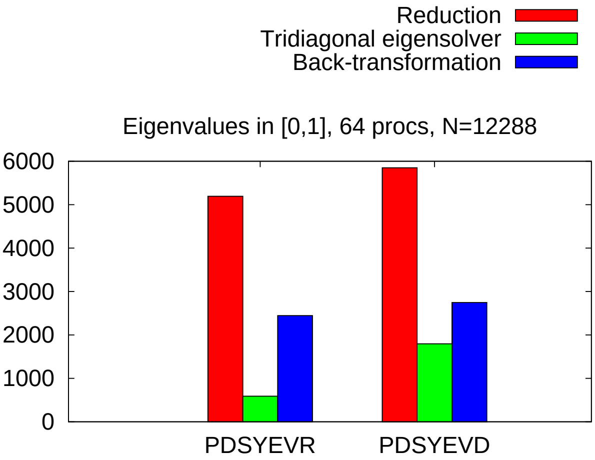
Todo

- Larger matrices with more processors
- Alternative MRRR versions
- Alternative reduction routines (not available yet)
- Other matrix spectra

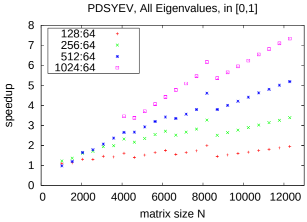
Thank you for your attention!

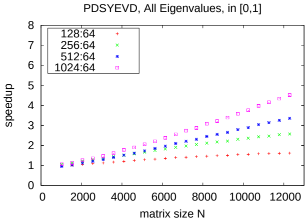
Questions?

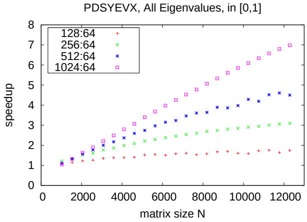
Comparison of the Three Computation Phases

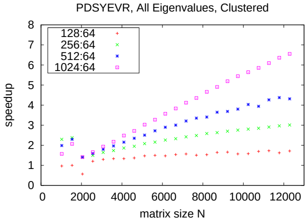


Speedups


PDSYEV, All Eigenvalues, in [0,1]
128:64
256:64
512:64
1024:64
speedup
matrix size N
0
1
2
3
4
5
6
7
8
0
2000
4000
6000
8000
10000
12000


PDSYEVD, All Eigenvalues, in [0,1]
128:64
256:64
512:64
1024:64
speedup
matrix size N
0
1
2
3
4
5
6
7
8
0
2000
4000
6000
8000
10000
12000


PDSYEVX, All Eigenvalues, in [0,1]
128:64
256:64
512:64
1024:64
speedup
matrix size N
0
1
2
3
4
5
6
7
8
0
2000
4000
6000
8000
10000
12000


PDSYEVR, All Eigenvalues, Clustered
128:64
256:64
512:64
1024:64
speedup
matrix size N
0
1
2
3
4
5
6
7
8
0
2000
4000
6000
8000
10000
12000

Elin Solberg
Department of Mathematical Sciences
University of Gothenburg, Sweden
solberg@student.gu.se