Verbundprojekt ELPA-AEO

http://elpa-aeo.mpcdf.mpg.de

Eigenwert-Löser für Petaflop-Anwendungen – Algorithmische Erweiterungen und Optimierungen

GEFÖRDERT VOM

Bundesministerium für Bildung und Forschung

BMBF Projekt 01IH15001
Feb 2016 - Jan 2019

Project Partners

Lehrstuhl für Angewandte Informatik
Prof. B. Lang (BUW)

Prof. M. Scheffler, Dr. Ch. Carbogno (FHI)

Dr. H. Lederer, Dr. A. Marek (MPCDF)

Lehrstuhl für Informatik mit Schwerpunkt
Wissenschaftliches Rechnen (TUM-SCCS)
Prof. H.-J. Bungartz, Prof. Th. Huckle

Lehrstuhl für Theoretische Chemie
Prof. K. Reuter, Dr. Ch. Scheurer (TUM-CH)

Eigensolvers for Symmetric Matrices
Fields of Relevance

Materials Science

Electron structure calculations

Technical and biological networks

Structural analyses

Structural Mechanics

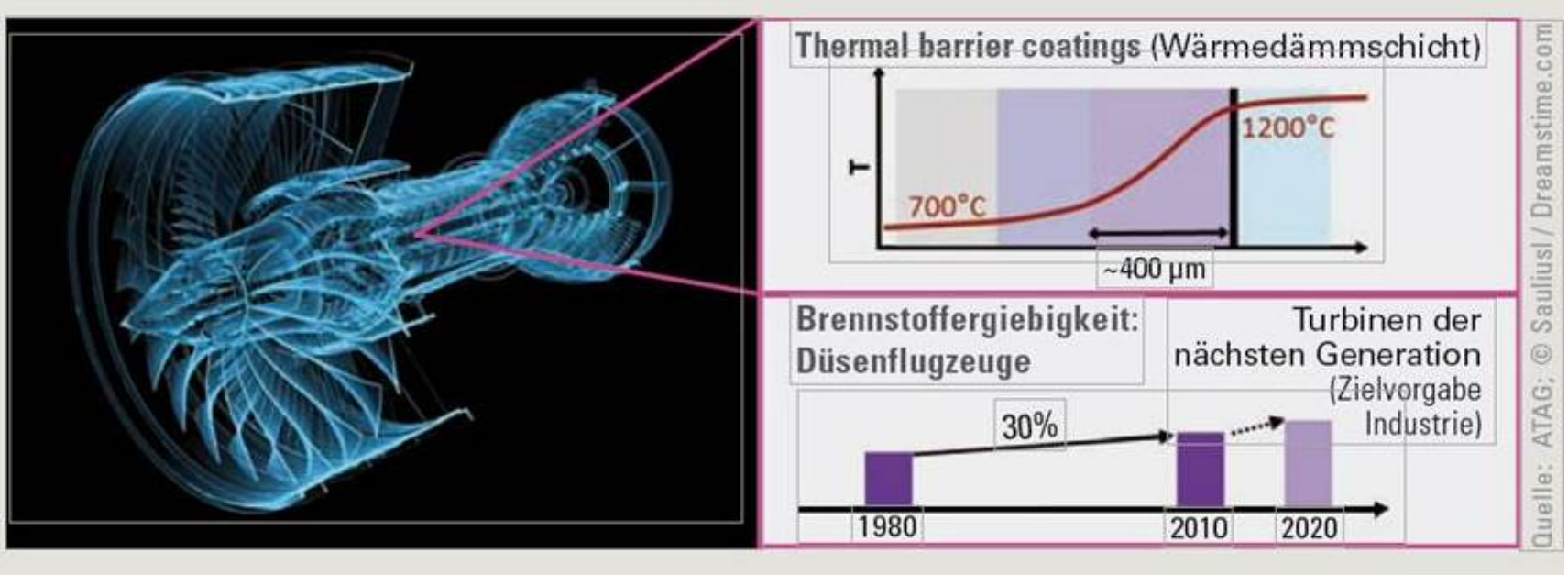
Building mechanics, statical calculations

Fluid Mechanics

Unsteady flows, complex turbulence models,
non-linear optimisation procedures for sound propagation

Relevance in Materials Science

- Large-scale, all-electron electronic structure theory
- Prediction of materials properties from the atomic scale on upwards, based only on the first principles of quantum mechanics.



Central task of electronic structure theory:
Solution of Schroedinger like eigenproblems

$$\hat{H}\Psi_m = E_m \Psi_m$$

ELPA-AEO

Stepwise Approach for the Eigenproblem

MPCDF Max Planck Computing & Data Facility

Solution of the generalized matrix eigenproblem generally proceeds in five steps:

(I) Transformation to a dense standard eigenproblem (e.g., by Cholesky decomposition of S),
$H_{KS}c_I = \varepsilon_I S c_I \rightarrow A q_A = \lambda q_A, \quad \lambda = \varepsilon_I$

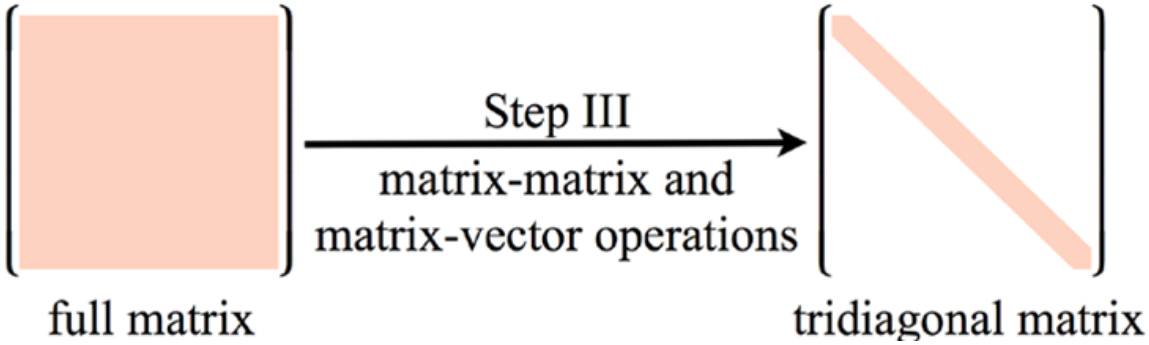
(II) Reduction to tridiagonal form, A -> T;

(III) Solution of the tridiagonal problem for k eigenvalues and -vectors, $T q_T = \lambda q_T$;

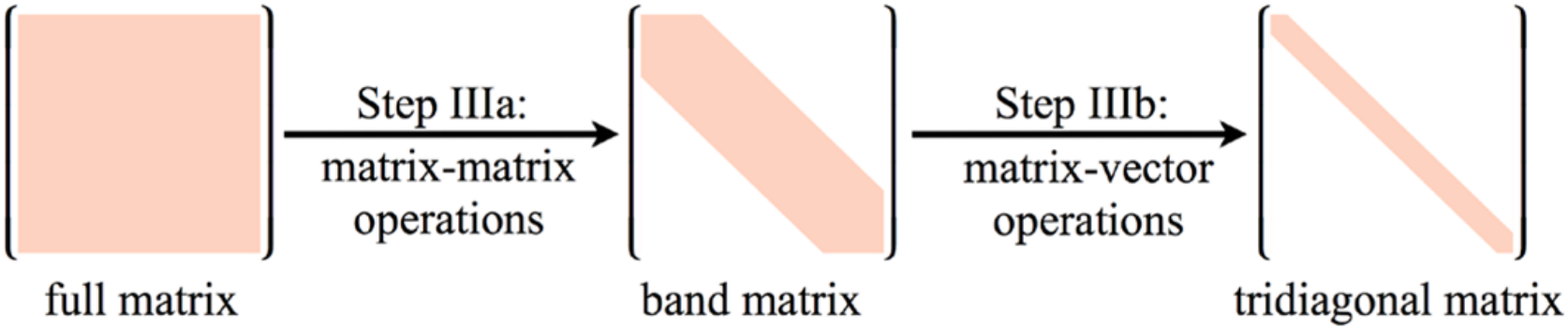
(IV) Back transformation of k eigenvectors to dense orthonormal form, $q_T \rightarrow q_A$;

(V) Back transformation to the original, non-orthonormal basis, $q_A \rightarrow c_I$.

ELPA1: one-step direct solver



ELPA2: two-step direct solver



Pros: *allows full use of matrix–matrix products and sparse matrix-vector products*
Cons*: gives rise to one extra back transformation step of the eigenvectors*

ELPA-AEO Goals

- Increasing the efficiency of supercomputer simulations for which the solution of the eigenproblem for dense and band-structured symmetric matrices is a significant contribution
- Enabling the addressing of even larger problems
- Reduction of the computational effort for the simulation
- Reduction of resource and energy usage for a given level of accuracy while keeping high software scalability

Improvements of time-to-solution and energy-to-solution

Work Packages

Algorithmic Developments	Lead: Univ. Wuppertal (AI)
Efficient parallelisation	Lead: TUM (SCCS)
Implementation, porting and optimisation	Lead: MPCDF
Autotuning	Lead: MPCDF
Applications I	Lead: Fritz-Haber-Institut, Theory Dep.

Context: Surfaces and solid-solid interfaces (e.g. thermo-electrics)

Applications II Lead: TUM (Theoretical Chemistry)

Context: solid-liquid and liquid-liquid Interfaces (e.g. photo-electro catalysis)

Algorithmic Developments

- Development of a direct eigensolver for generalized eigenproblems with band structure
- Studies of the suitability (and potentially development) of an iterative eigensolver for generalized eigenproblems with band structure
- Development of numerical techniques for the reduction of the effort at reduced accuracy requirements and for the generation of an approximative band structure
- Development of strategies for the reduction of the effort for sequences of eigenproblems
- Identification of highly efficient variants of the transformation to the standard eigenproblem
- Application specific continued development of methods for the reduction of the approximative band structure (localisation of base set)
- Development and adaption of algorithms for preconditioning and stabilisation for the non-linear problem (SCF)

ELPA-AEO

MPCDF Max Planck Computing & Data Facility

Efficient Parallelisation

Efficient parallel realisation of algorithms and strategies developed by WP1, especially the back-transformation of the eigenvectos for band-structured problems and conservation of orthogonality of many vectors in the iterative eigensolver.

Implementation, Porting and Optimisation

Implementations and optimisations of the results of WP2,
porting and optimisation of ELPA routines with respect to new architectures,
consideration of feedback from WP5 and WP6 for quality, stability and overall efficiency

Autotuning

Concept and realization of a monitoring tool for the behaviour of the different used components; automatic ELPA autotuning; definition and implementation of an API for the reporting to the calling main programme as well as for manually set requirements; identification ot the necessary accuracy limits for applications I and II;
developments of semi-manual tuning strategies at application level

ELPA-AEO

MPCDF Max Planck Computing & Data Facility

Applications I + II

- Provisioning of large matrix sizes and application cases related to
  -> surfaces and solid-solid interfaces (e.g. thermo electrics)
  -> solid-liquid and liquid-liquid interfaces (e.g. photo-electro-catalysis)

- Integration of new algorithms into the SCF *(Self Consistent Field)* - and ab-initio molecular dynamics methods in the simulation package FHI-aims

- Monitoring of quality, stability and total efficiency of the newly developed algorithms in both application contexts

- Tier-0 simulations with scientific relevance

ELPA-AEO

Autotuning

MPCDF Max Planck Computing & Data Facility

- Monitoring tools for the behaviour of the different components
- Automatic ELPA autotuning
- API for reporting to the calling main program and for manual presettings according to application requirements
- Identification of the required precision limits for calculations
- Semi-manual tuning strategies at application level

ELPA-AEO

Automatic ELPA Autotuning

MPCDF Max Planck Computing & Data Facility

For a given problem hardware configuration:
Facilitating the selection of the most efficient procedure for the solution of the eigenproblem

For a given problem (matrix size and number of eigenvalues needed) comparison of the different procedures (elpa1 or elpa2) and identification of the most efficient one

For elpa2 (more complex than elpa1) : optimization of the selection of the block size and the different compute-intenisve kernels. In case of availability of GPUs, inclusion into selection criteria.

Milestones after 24 months

M24-1:
Implementation of the generalized band eigensolver with calculation of eigenvectors realized

M24-2:
Implementation of the monitoring tool for autotuning realized for selected routines

M24-3:
Variants for reduced precision requirements implemented for selected routines

Results

New Algorithms

Direct eigensolver for generalized eigenproblem with calculation of eigen vectors

- New algorithm already developed
  (Univ. Wuppertal, AI)

- Implementation of parallel version nearly ready
  (TUM-SCCS)

Results

Monitoring

- Implementation of the monitoring tool for autotuning realized for selected routines

- Monitoring: detailed timing measurements (both exclusive and inclusive) realized for the major routines (elpa1 solver, elpa2 solver, elpa2 optimized kernels)

- Calling program can
- enable/disable timing measurements
- query timing results, number of calls of a subroutine and performance in GigaFlop/s
- print timing results to stdout

Results

Autotuning

Based on monitoring infrastructure

Main program options: enable/disable (default: disable)

In case of enable:

When elpa is called multiple times within an SCF iteration, different elpa solvers (elpa 1 and elpa 2) and different tuning options for elpa 2 (kernel) are called consecutively and the timings recorded. After measuring all available options, the best performing one is used for all subsequent calls .

User guided autotuning:

User can preselect settings (e.g. elpa2) which is then fixed and other settings (elpa 1) are not considered.

Results

Variants for reduced precision requirements

Default for calculations: **double precision**

Additionally implemented and provided to users for testing: **single precision versions** of elpa - both solvers and transformation routines for generalized eigenvalue (EVP)

Users can **specify** in their main program **which variant to use**.

Results

Porting to and optimization for new architectures

- Intel Xeon Phi KNL: ELPA ported and published
  (including basic AVX512 optimization)

- Intel Xeon SkyLake: extended AVX512 optimization publ.

- GPGPUs: NVIDIA Tesla GPGPUs: K20, K40, K80, P100:

  ELPA ported for usage on Intel Xeon hosts via PCIe
  ELPA ported for usage on Open Power via NVLINK1
  Minsky – Node architecture:
  2 CPUs IBM Power8 + 4 GPUs NVIDIA P100

New ELPA Releases

ELPA-Release from Nov 2016

- Support for Intel Xeon KNL
- Support for NVIDIA Tesla GPGPUs via PCIe in X86 host CPUs
- 32 bit versions of elpa1 and elpa2 for X86 and GPUs

ELPA-Release from May 2017

- Additional new interface published
  (enabling communication between elpa and main calling program);
  previous interface still supported (without the new functionalities)
- Extended AVX512 instructions supported for Intel Xeon

ELPA-Release from Nov 2017

- Autotuning supported for selected routines

ELPA runs and performance

- **Intel Xeon KNL (Theta, ANL):**
  Large runs and scalability

- **GPU systems**

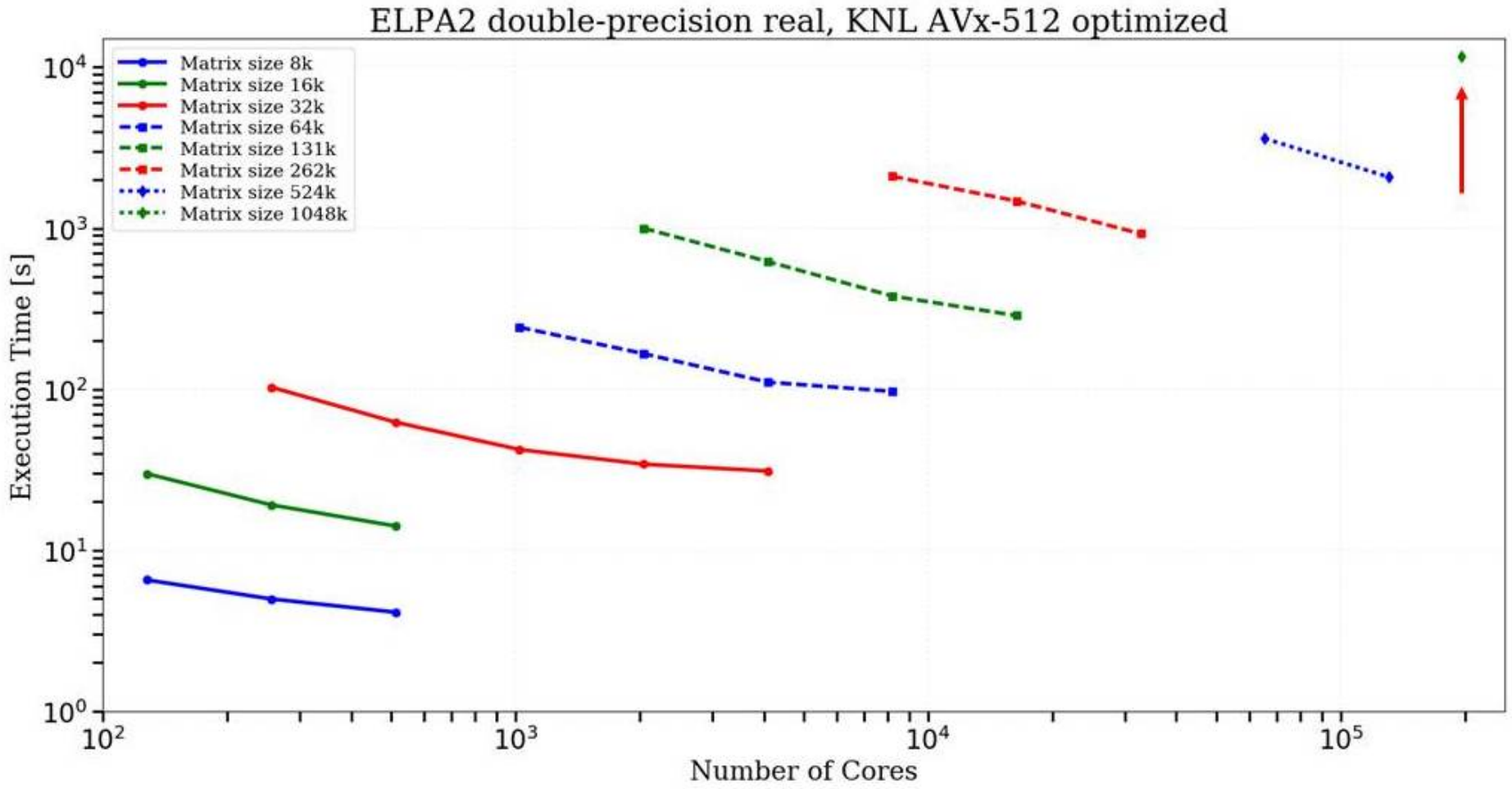
a) K40 with X86 host via PCIe: 1 +2 nodes
b) P100 with Power8 hosts via NVLINK1 (Minsky)
c) GTX1080 with X86 host via PCIe

- **Intel Xeon E5 2680 v2**

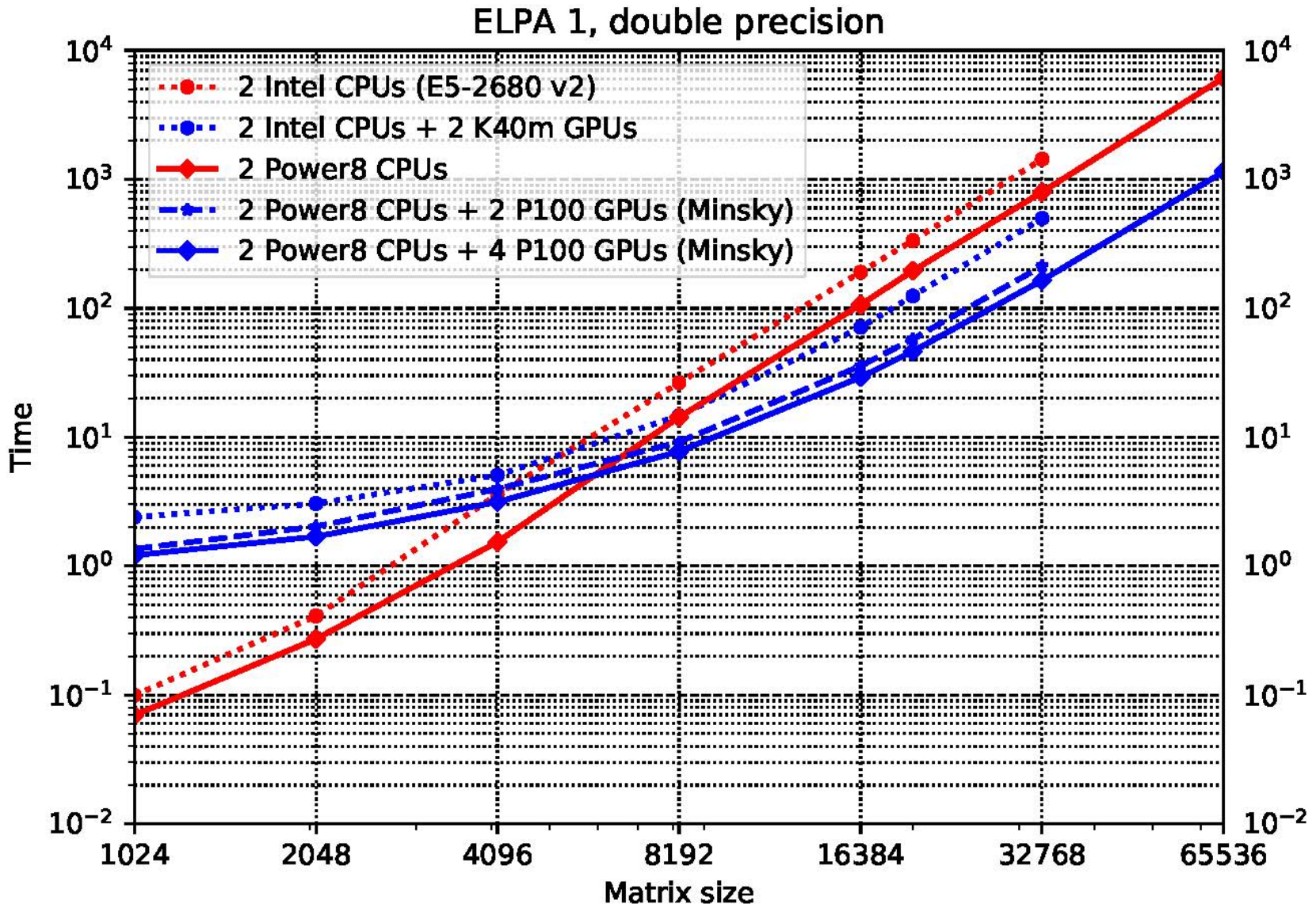
a) Double precision (2017 vs 2013)
b) Single precision (2017)

ELPA scalability: runs up to 200 k-cores on Intel Xeon Phi KNL

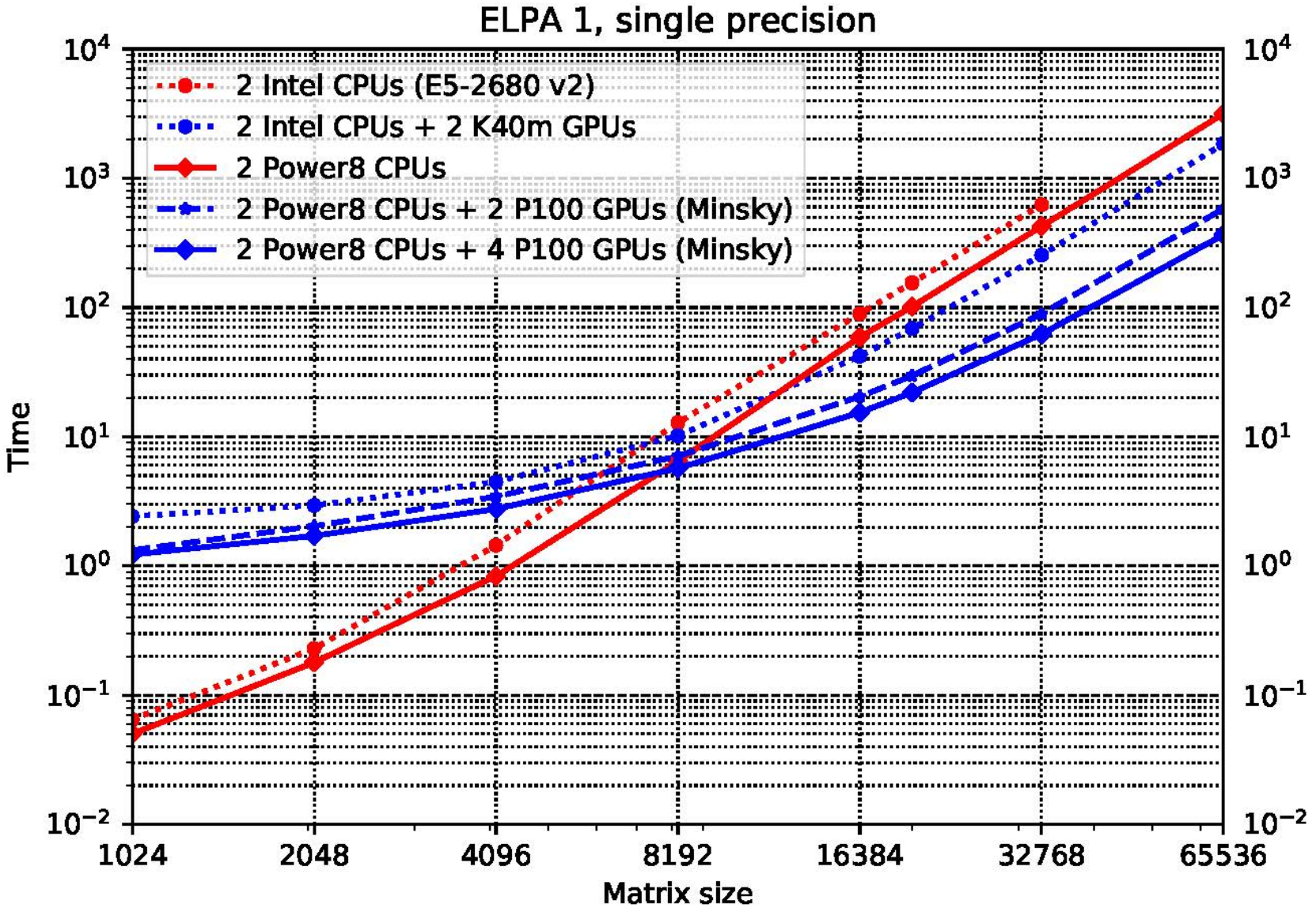
(Theta at Argonne National Lab; courtesy of Vazquez-Mayagoit Alvaro, ANL)



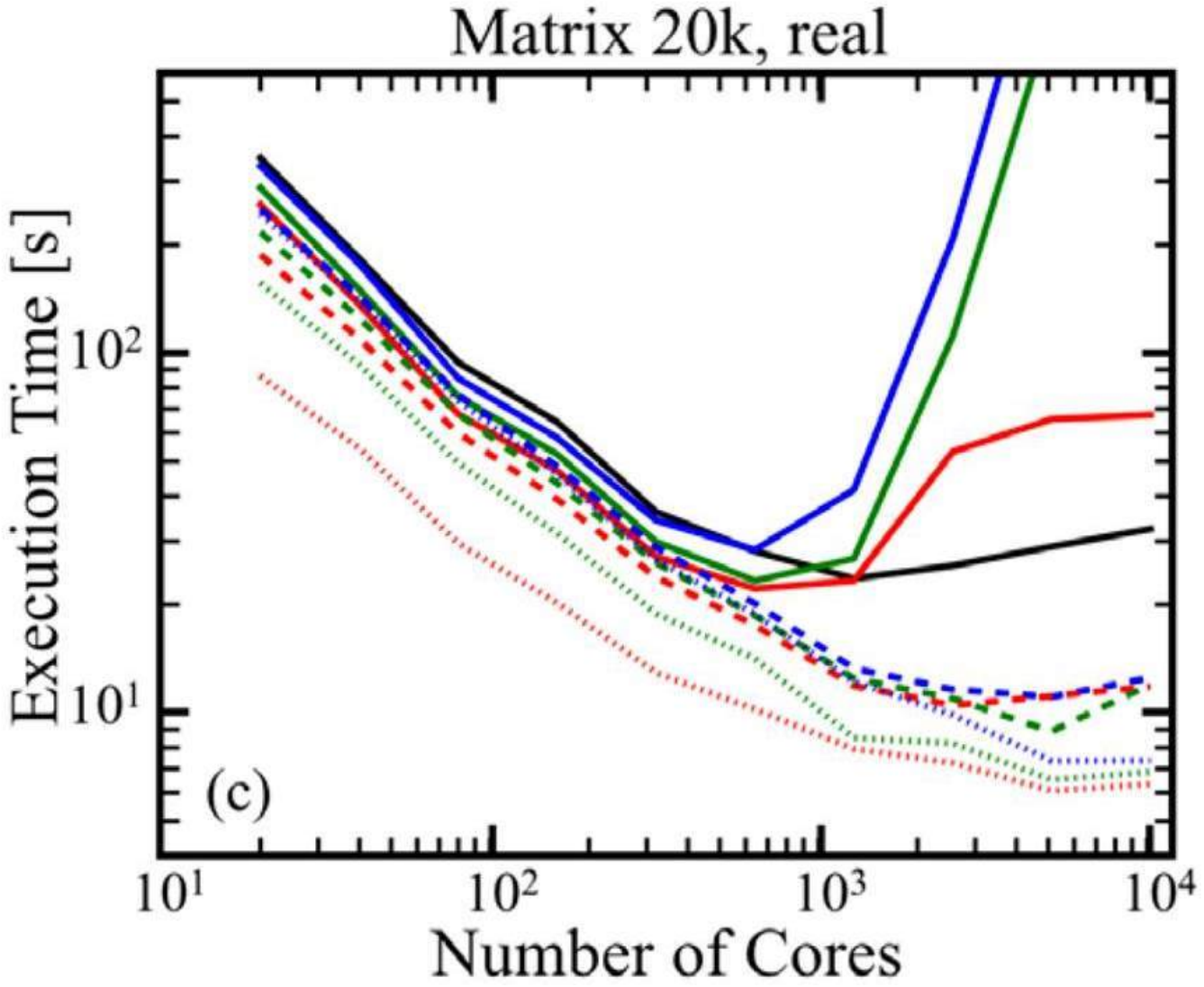
ELPA on Kepler GPUs (K40 and P100)



ELPA on Kepler GPUs (K40 and P100)

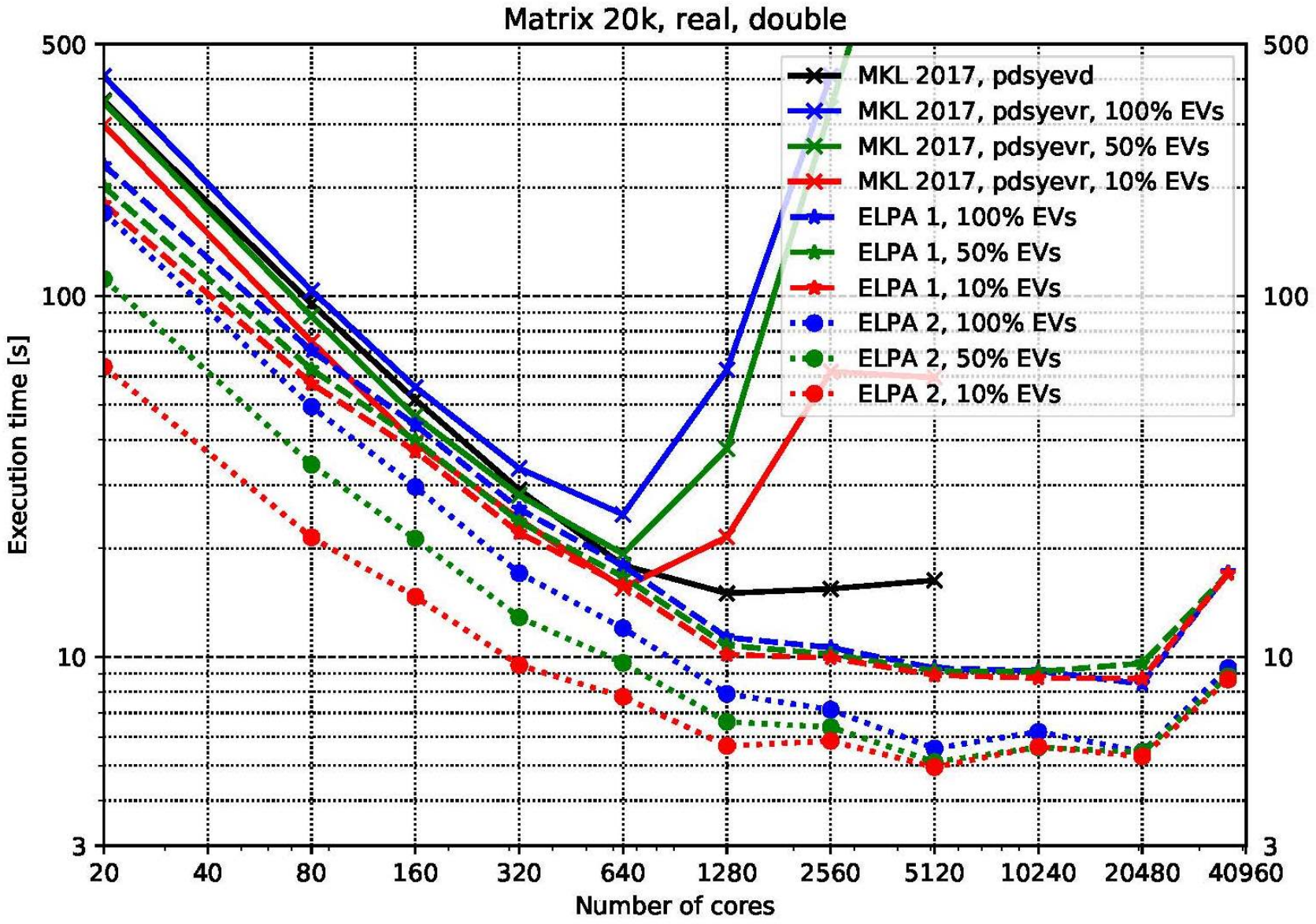


ELPA on Intel Xeon (E5-2680v2) - 2013

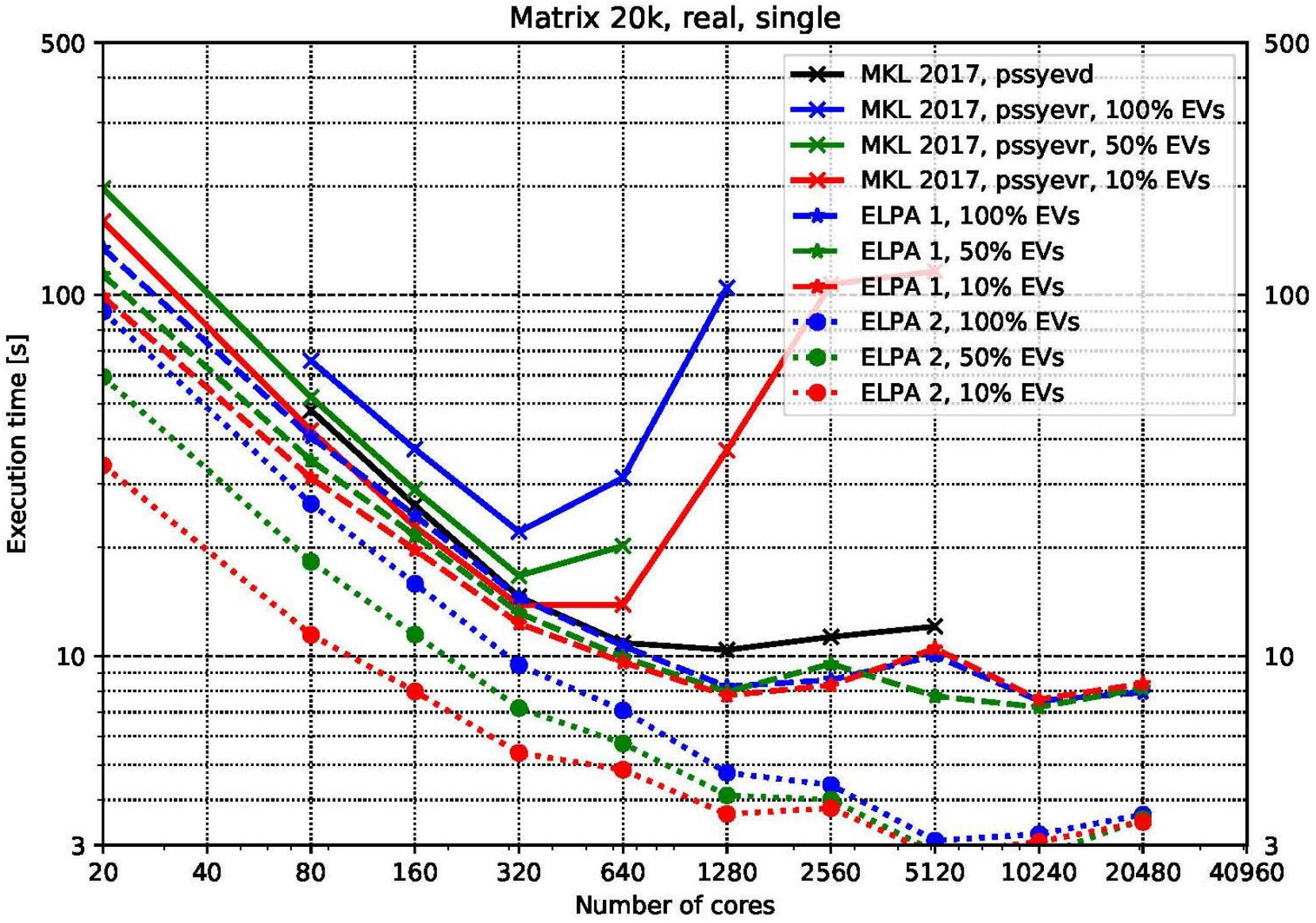


Journal of Physics: Condensed Matter Vol. 26, No. 21 (2014)

ELPA on Intel Xeon (E5-2680v2) - 2017



ELPA on Intel Xeon (E5-2680v2) - 2017



ELPA Performance Comparison

System: Hydra (node: 2 x Intel Xeon E5-2680v2, 20 cores)
Matrix size: 20 000, real
Solver: ELPA2, 100% EVs

2013, double precision on 80 cores: ~ **75 s**

2017, double precision on 80 cores: ~ **50 s**
improvement factor (2017 double)/(2014 double): **~ 1.5**

2017, single precision on 80 cores: ~ 30 s
improvement factor (2017 single)/(2017 double): ~ 1.8
improvement factor (2017 single)/(2014 double): ~ 2.7