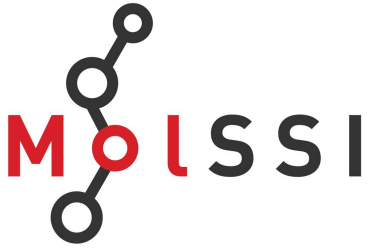
Large-Scale Electronic Structure Methods (ELPA and ELSI)

Victor Yu

Materials Science Division, Argonne National Laboratory

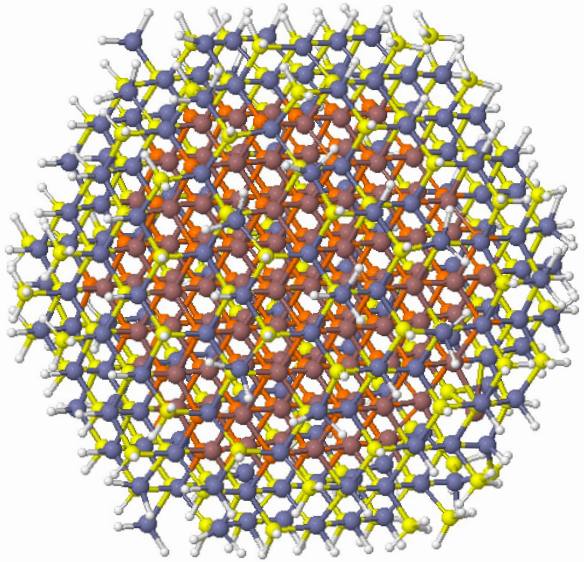
Research in this presentation conducted at Duke University,
funded by NSF awards 1450280 (ELSI) and 1547580 (MolSSI)



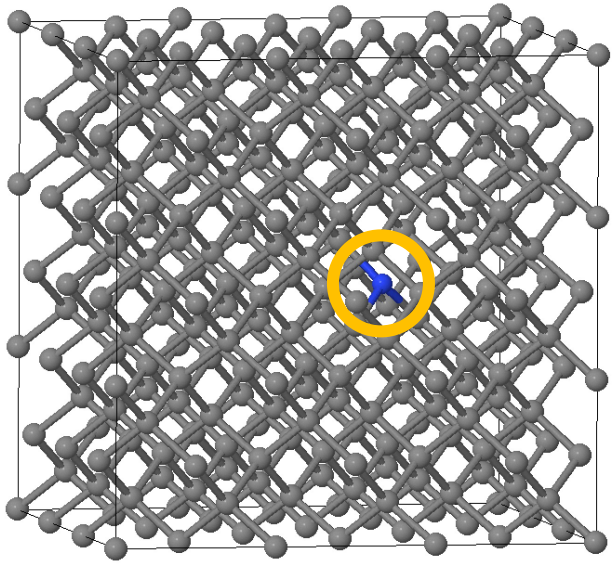
This talk is about ...

Applying Kohn-Sham density-functional theory (KS-DFT) to large systems

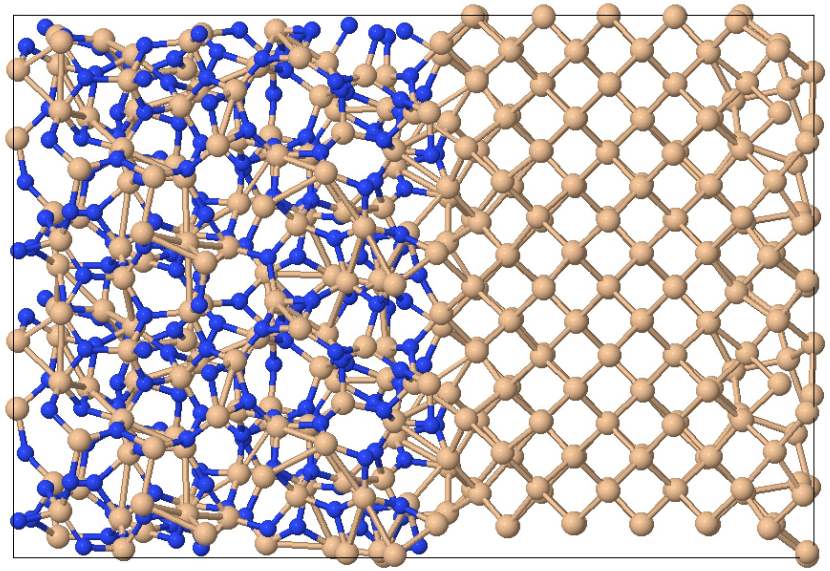
Nanostructure



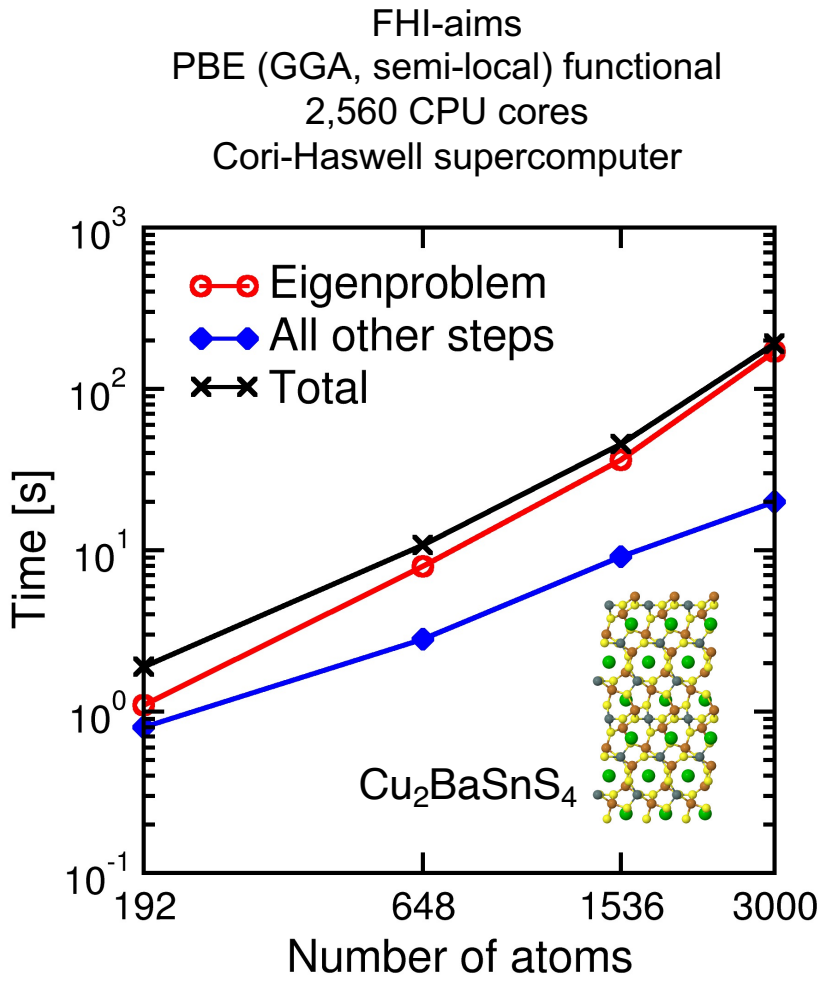
Defect



Interface



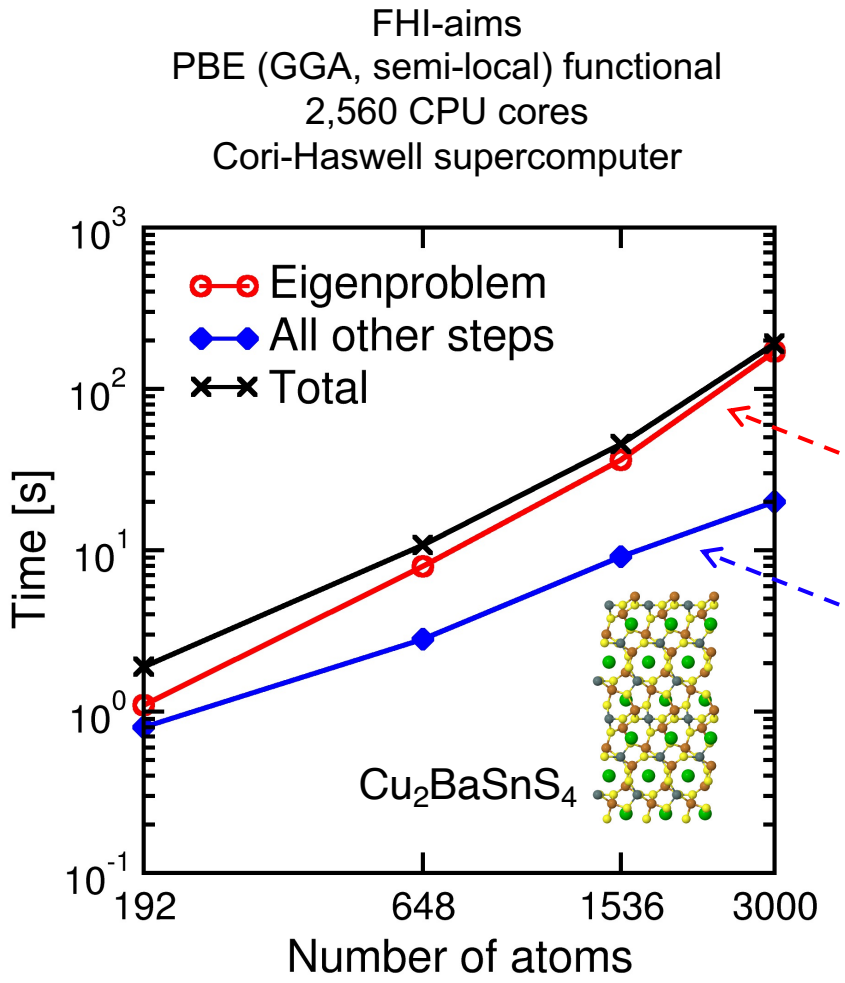
The “cubic wall” in Kohn-Sham density-functional theory



Key computational steps in KS-DFT
(with semi-local exchange-correlation functionals)

- Integrate KS Hamiltonian
- Diagonalize KS Hamiltonian (eigensolver)
- Compute electron density
- Evaluate potential from density

The “cubic wall” in Kohn-Sham density-functional theory



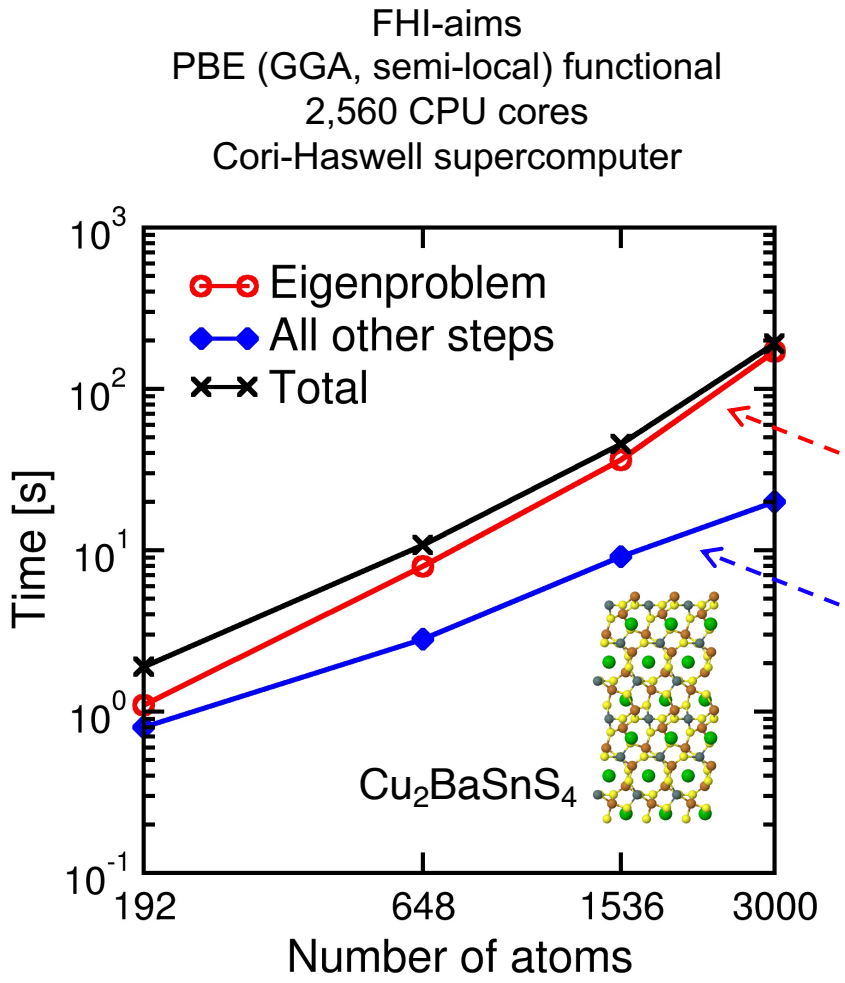
Key computational steps in KS-DFT
(with semi-local exchange-correlation functionals)

- Integrate KS Hamiltonian
- Diagonalize KS Hamiltonian (eigensolver)
- Compute electron density
- Evaluate potential from density

Computational cost of a dense eigensolver scales as $O(N^3)$ with respect to the system size N

The other steps can be formulated as O(N)
(previous talk by Dr. Ville Havu)

The “cubic wall” in Kohn-Sham density-functional theory



Key computational steps in KS-DFT
(with semi-local exchange-correlation functionals)

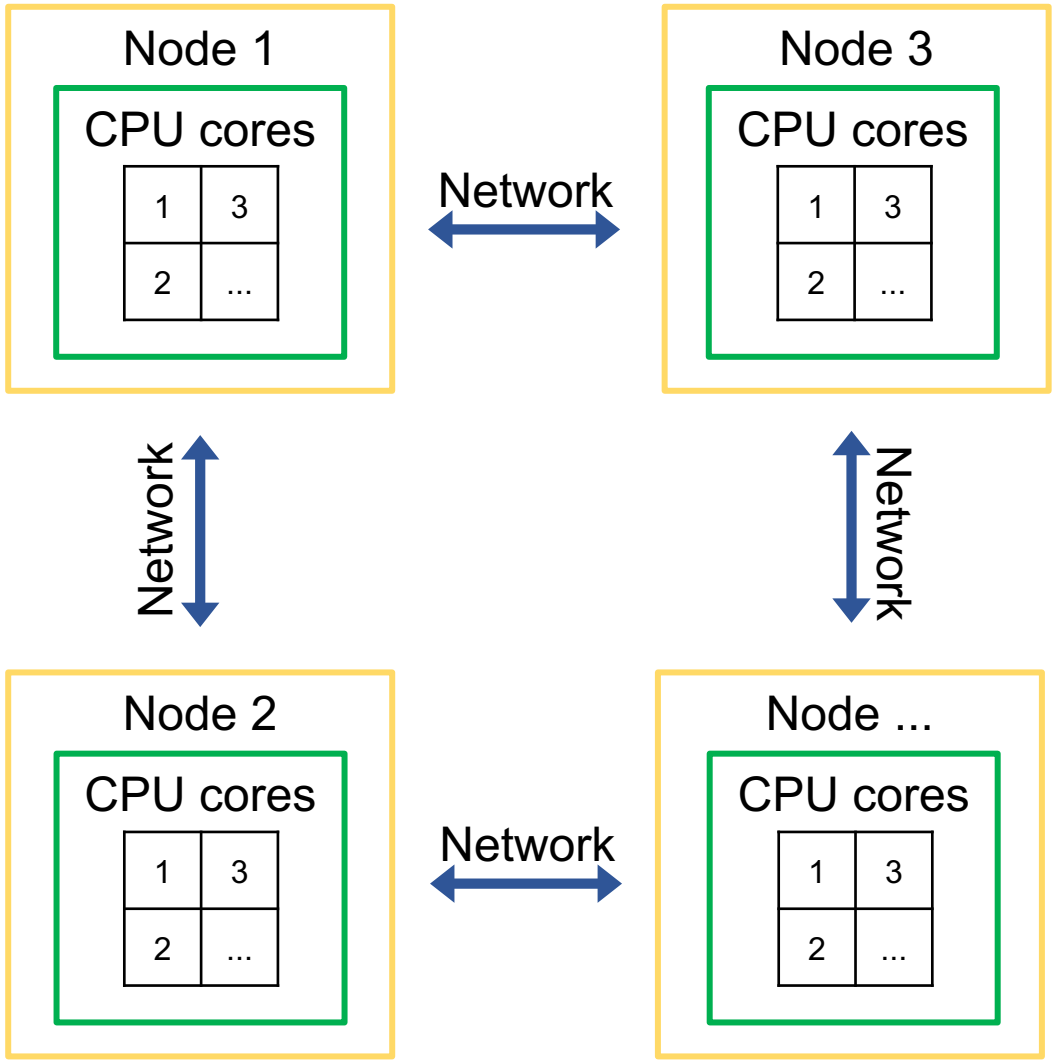
- Integrate KS Hamiltonian
- Diagonalize KS Hamiltonian (eigensolver)
- Compute electron density
- Evaluate potential from density

Computational cost of a dense eigensolver scales as $O(N^3)$ with respect to the system size N

The other steps can be formulated as $O(N)$
(previous talk by Dr. Ville Havu)

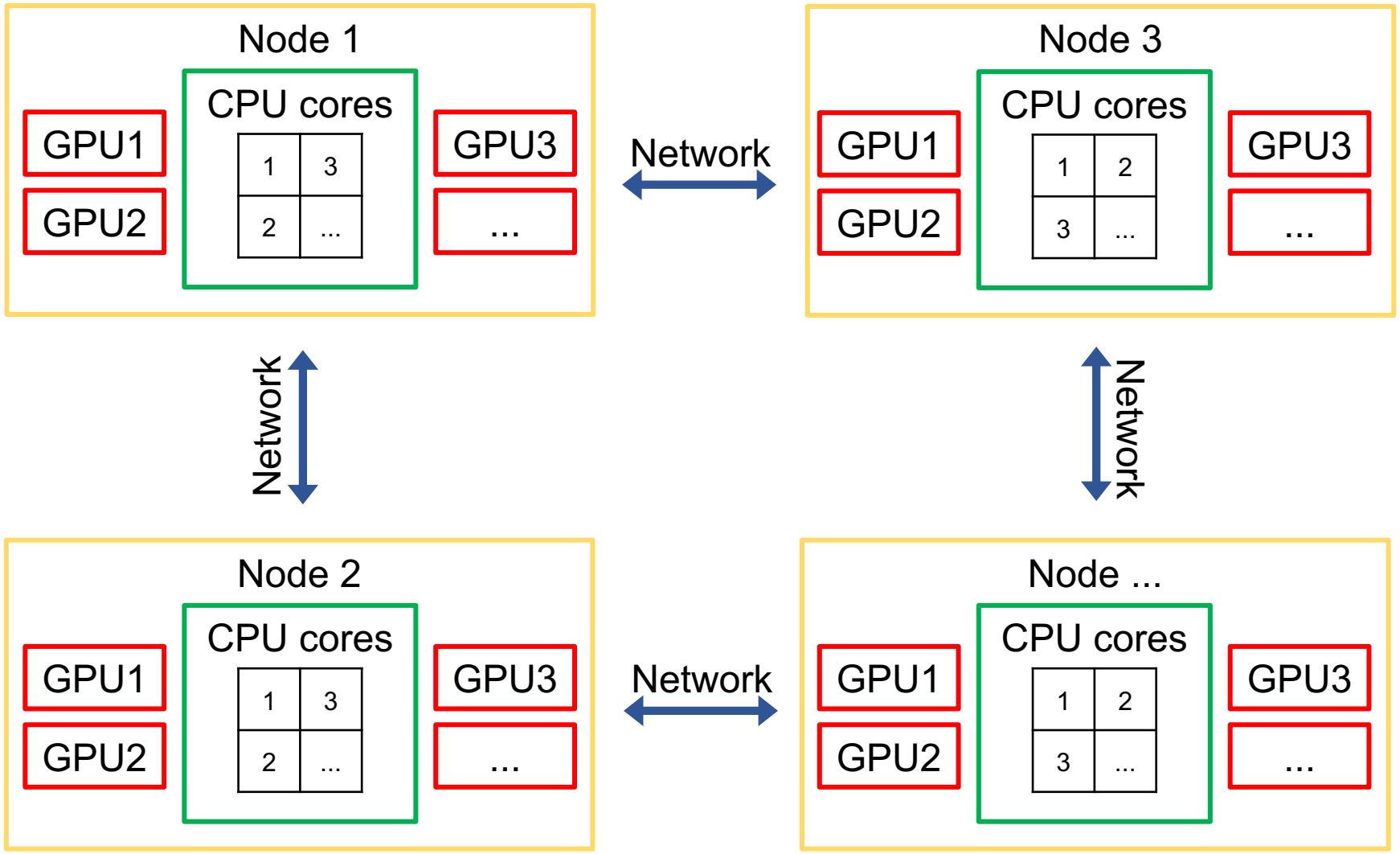
How to speed up the diagonalization step?

High-performance computers



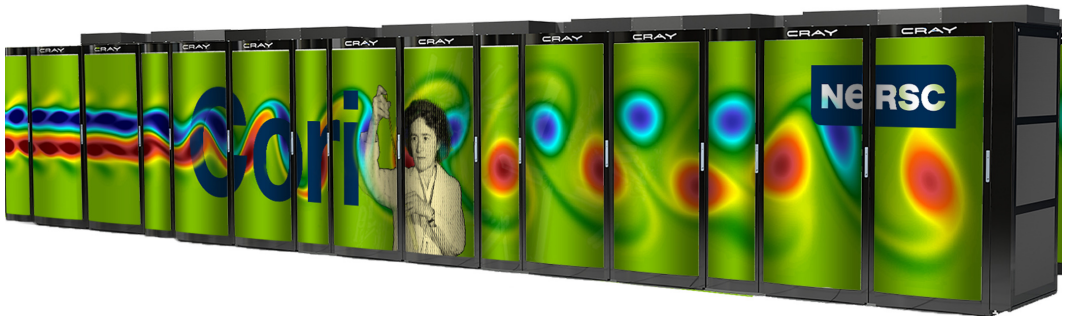
Large number (100~10,000) of CPU nodes connected together

High-performance computers



Large number (100~10,000) of CPU+accelerator nodes connected together

High-performance computers



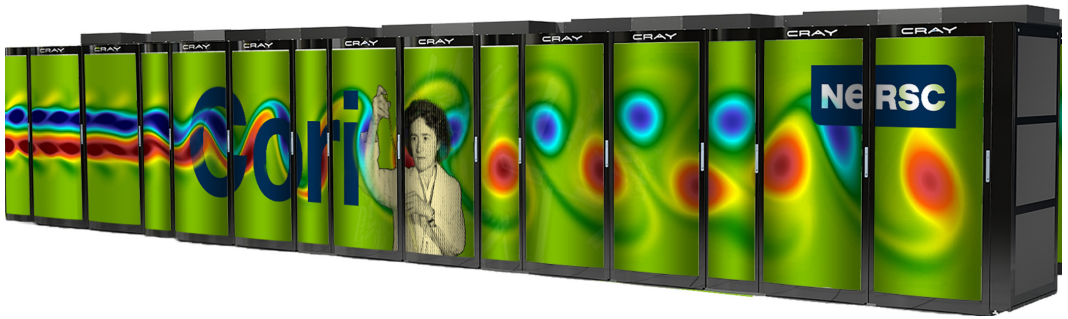
https://www.nersc.gov/cori

Supercomputer “Cori”
(world’s no. 30)

2,388 Intel Haswell CPU nodes
9,688 Intel Knights Landing CPU nodes
27.8 PFLOP/s

1 PFLOP/s = 10^{15} floating point operations per second

High-performance computers



https://www.nersc.gov/cori

http://www.olcf.ornl.gov/olcf-resources/compute-systems/summit

Supercomputer “Cori”
(world’s no. 30)

2,388 Intel Haswell CPU nodes
9,688 Intel Knights Landing CPU nodes
27.8 PFLOP/s

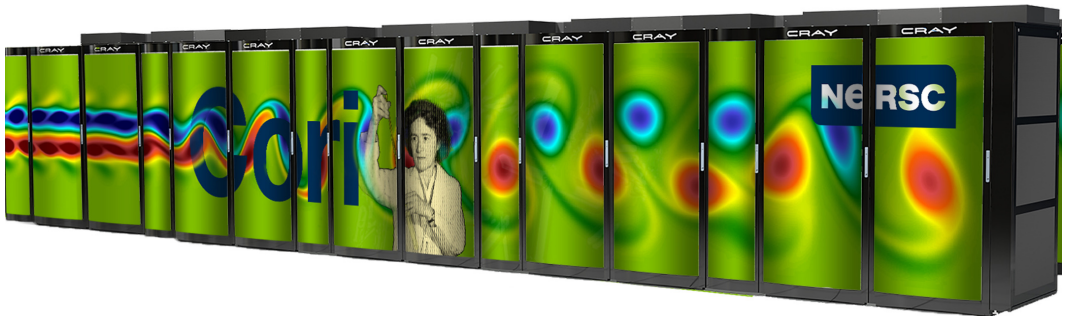
Supercomputer “Summit”
(world’s no. 2)

4,608 nodes
6 NVIDIA V100 GPUs per node
200 PFLOP/s

1 PFLOP/s = 10^{15} floating point operations per second

High-performance computers

Run FHI-aims in parallel: mpirun -n 1000 aims.scalapack.mpi.x



https://www.nersc.gov/cori

http://www.olcf.ornl.gov/olcf-resources/compute-systems/summit

Supercomputer “Cori”
(world’s no. 30)

2,388 Intel Haswell CPU nodes
9,688 Intel Knights Landing CPU nodes
27.8 PFLOP/s

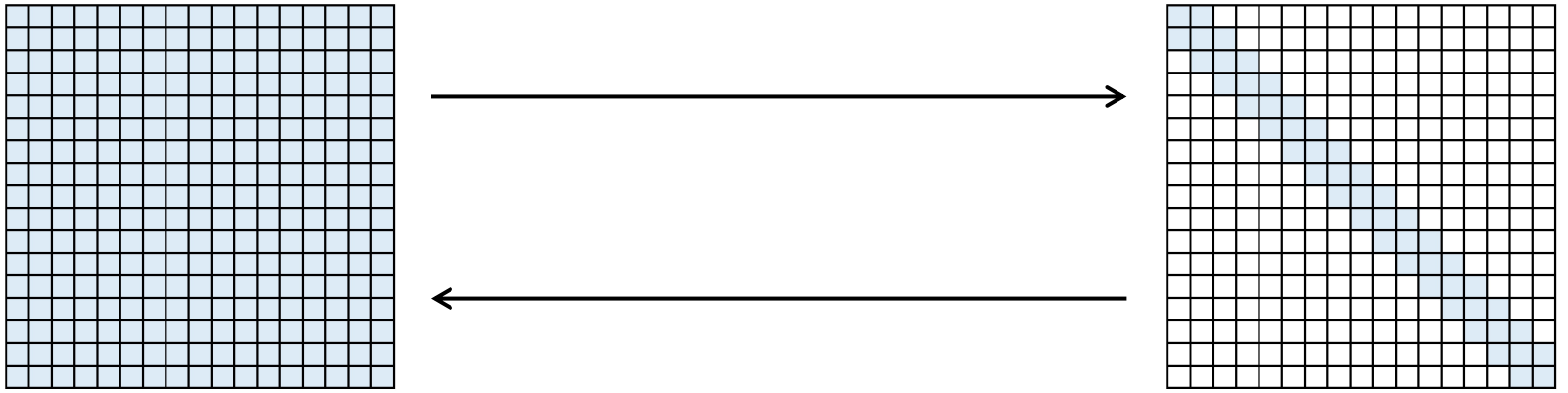
Supercomputer “Summit”
(world’s no. 2)

4,608 nodes
6 NVIDIA V100 GPUs per node
200 PFLOP/s

1 PFLOP/s = 10^{15} floating point operations per second

ELPA, a massively parallel eigensolver

Massively parallel eigensolver ELPA



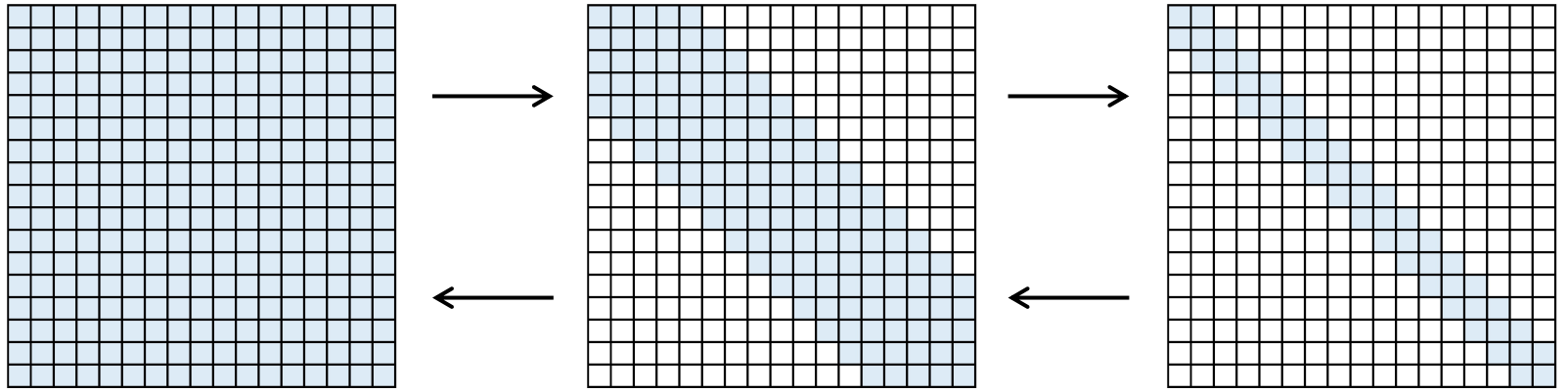
One-stage diagonalization (textbook)

1) Full matrix A → tridiagonal matrix T
2) Solve tridiagonal matrix T
3) Tridiagonal eigenvectors → full

Mostly matrix-vector multiplications in step1

Auckenthaler et al., Parallel Comput. **37** (2011), 783-794
Marek et al., J. Phys. Condens. Matter **26** (2014), 213201

Massively parallel eigensolver ELPA



One-stage diagonalization (textbook)

1) Full matrix A → tridiagonal matrix T
2) Solve tridiagonal matrix T
3) Tridiagonal eigenvectors → full

Mostly matrix-vector multiplications in step1

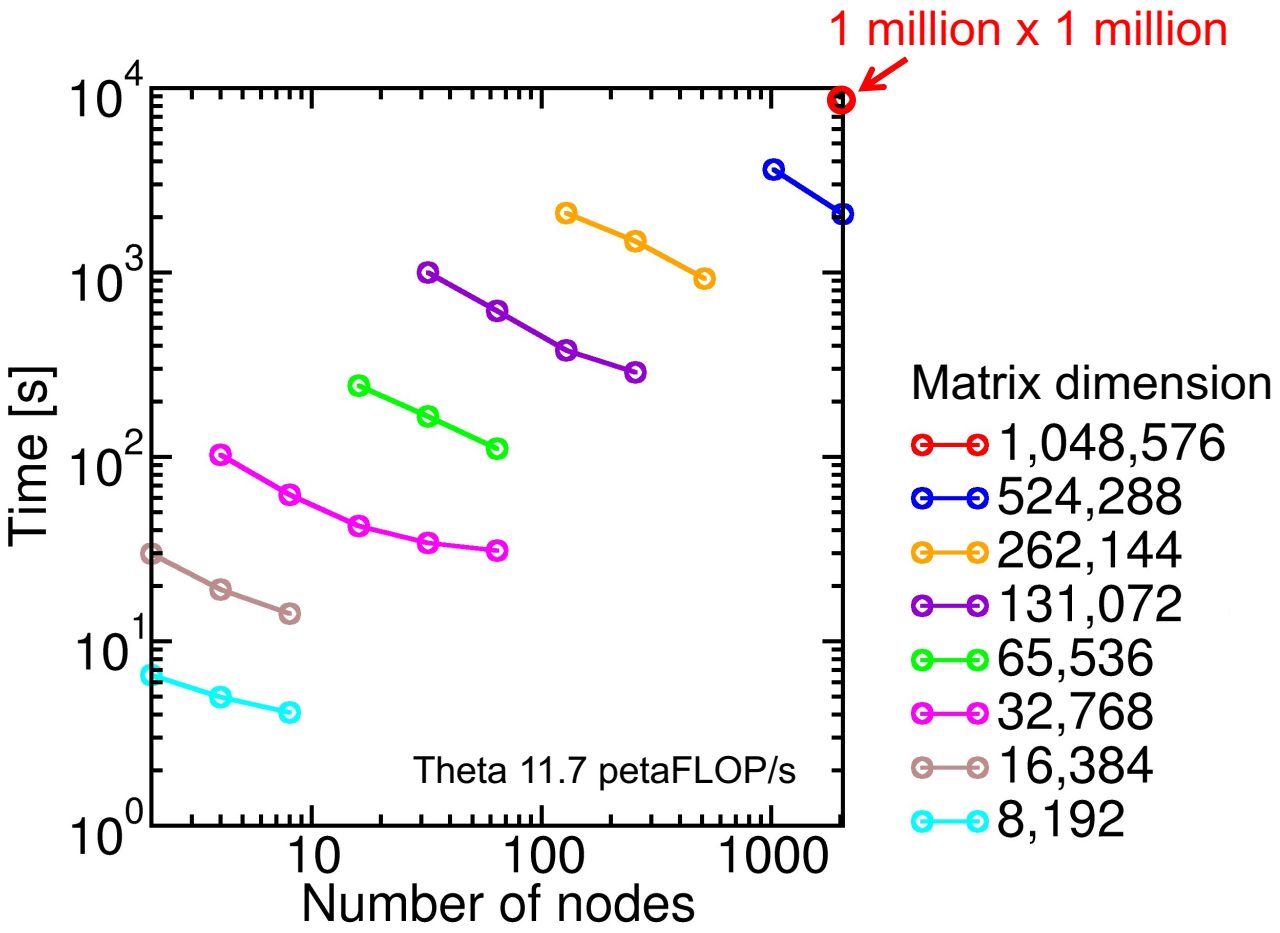
Two-stage diagonalization

1) Full matrix A → banded matrix B
2) Banded matrix B → tridiagonal matrix T
3) Solve tridiagonal T
4) Tridiagonal eigenvectors → banded
5) Banded eigenvectors → full

More matrix-matrix operations in step1

Auckenthaler et al., Parallel Comput. **37** (2011), 783-794
Marek et al., J. Phys. Condens. Matter **26** (2014), 213201

Massively parallel eigensolver ELPA



1 Theta node = 64 CPU cores
Data from Dr. Álvaro Vázquez-Mayagoitia, Argonne National Laboratory

https://elpa.mpcdf.mpg.de

- Massively parallel (efficient for 10k+ CPU cores)
- Used in more than 20 DFT software packages

Can we do better?

Bypass diagonalization?

Recap: Kohn-Sham density-functional theory (KS-DFT)

Kohn-Sham equations

$$\hat{h}_{KS}\psi_l = \epsilon_l \psi_l$$

Basis set expansion

$$\psi_l(\mathbf{r}) = \sum_j c_{lj}\varphi_j(\mathbf{r})$$

φ_j: Basis functions

c_{lj}: Expansion coefficients

Generalized eigenproblem

$$\mathbf{HC} = \mathbf{SC\Sigma}$$

Recap: Kohn-Sham density-functional theory (KS-DFT)

Kohn-Sham equations

$$\hat{h}_{KS}\psi_l = \epsilon_l \psi_l$$

Basis set expansion

$$\psi_l(\mathbf{r}) = \sum_j c_{lj}\varphi_j(\mathbf{r})$$

φ_j: Basis functions

c_{lj}: Expansion coefficients

Generalized eigenproblem

$$\mathbf{HC} = \mathbf{SC\Sigma}$$

$$\mathbf{n}(\mathbf{r}) = \sum_l f_l |\psi_l(\mathbf{r})|^2$$

Explicitly solve eigenproblem
"diagonalization"

Recap: Kohn-Sham density-functional theory (KS-DFT)

Kohn-Sham equations

$$\hat{h}_{KS}\psi_l = \epsilon_l \psi_l$$

Basis set expansion

$$\psi_l(\mathbf{r}) = \sum_j c_{lj}\varphi_j(\mathbf{r})$$

φ_j: Basis functions

c_{lj}: Expansion coefficients

Generalized eigenproblem

$$\mathbf{HC} = \mathbf{SC\Sigma}$$

$$\mathbf{n}(\mathbf{r}) = \sum_l f_l |\psi_l(\mathbf{r})|^2$$

Explicitly solve eigenproblem "diagonalization"

$$\mathbf{n}(\mathbf{r}) = \sum_{i,j} \varphi_i^*(\mathbf{r}) p_{ij} \varphi_j(\mathbf{r})$$

Density matrix p_{ij} can be computed from $\mathbf{C}$ and $\mathbf{\Sigma}$, or can be constructed as a function of $\mathbf{H}$ and $\mathbf{S}$

Eigensolvers and density matrix solvers

(Sca)LAPACK
Tridiagonalization
$O(N^3)$

EigenExa
Penta-diagonalization
$O(N^3)$

MAGMA
Tridiagonalization with GPU
$O(N^3)$

ELPA
Two-stage tridiagonalization optionally with GPU
$O(N^3)$

Direct eigensolvers

Eigensolvers and density matrix solvers

(Sca)LAPACK Tridiagonalization $O(N^3)$	EigenExa Penta-diagonalization $O(N^3)$
MAGMA Tridiagonalization with GPU $O(N^3)$	FEAST Contour integral $\leq O(N^3)$
ELPA Two-stage tridiagonalization optionally with GPU $O(N^3)$	SLEPc Shift-and-invert plus spectrum slicing $\leq O(N^3)$

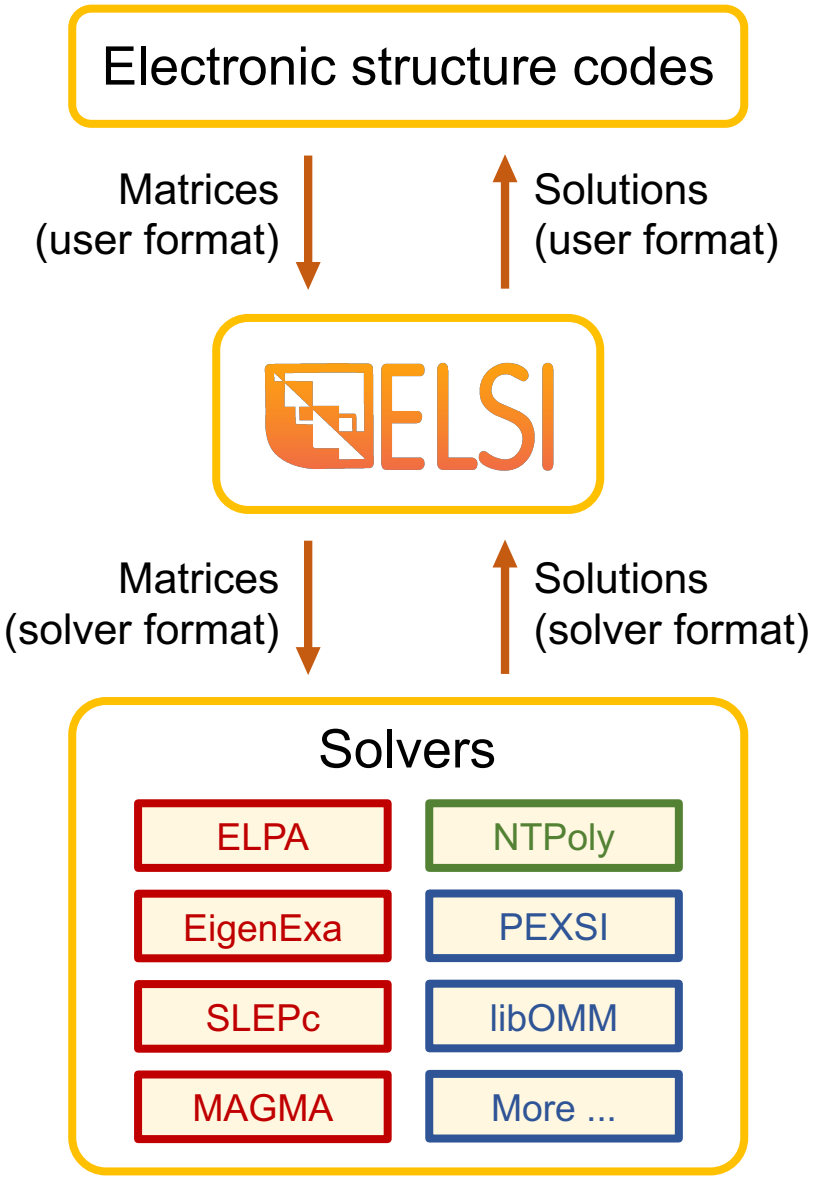
Direct eigensolvers
Iterative eigensolvers

Eigensolvers and density matrix solvers

(Sca)LAPACK Tridiagonalization $O(N^3)$	EigenExa Penta-diagonalization $O(N^3)$	libOMM Orbital minimization method $O(N^3)$
MAGMA Tridiagonalization with GPU $O(N^3)$	FEAST Contour integral $\leq O(N^3)$	NTPoly Density matrix purification $O(N)$
ELPA Two-stage tridiagonalization optionally with GPU $O(N^3)$	SLEPc Shift-and-invert plus spectrum slicing $\leq O(N^3)$	PEXSI Pole expansion and selected inversion $\leq O(N^2)$

Direct eigensolvers
Iterative eigensolvers
Density matrix solvers

ELSI: connecting electronic structure codes and solvers



- Unified interface for various solvers
- Parallel matrix (inter-)conversion
- Multi-level parallelization
- Reliable default parameters
- Recommendation of optimal solver

Yu et al., Comput. Phys. Commun. **222** (2018), 267-285
Yu et al., Comput. Phys. Commun. **256** (2020), 107459

ELSI used in the electronic structure community

All-electron KS-DFT
Numeric atom-centered orbitals
Sparse matrices

Blum et al., Comput. Phys. Commun. **180** (2009), 2175-2196

DGDFT

Pseudopotential KS-DFT
Adaptive local basis
Standard eigenproblems

Hu et al., J. Chem. Phys. **143** (2015), 124110

Pseudopotential KS-DFT
Pseudo atom-centered orbitals
More sparse matrices

García et al., J. Chem. Phys. **152** (2020), 204108

DFTB+

Semi-empirical tight-binding
Slater-type orbitals
Highly sparse matrices

Hourahine et al., J. Chem. Phys. **152** (2020), 124101

Pole expansion and selected inversion (PEXSI)

$$\mathbf{P} = \sum_{l} \mathrm{Im}\left(\frac{w_l}{\mathbf{H} - (z_l + \mu)\mathbf{S}}\right)$$

$\mathbf{P}$: Density matrix
$\mathbf{H}$: Hamiltonian matrix
$\mathbf{S}$: Overlap matrix
$\{z_l, w_l\}$: Poles
μ: Chemical potential

- Only compute selected elements of $(\mathbf{H} - (z_l + \mu)\mathbf{S})^{-1}$
- Computational cost
  - 1D system: O(N)
  - 2D system: $O(N^{1.5})$
  - 3D system: $O(N^2)$
- Generally applicable to insulating, semi-conducting, and metallic systems
- Poles evaluated independently in parallel (distributed across MPI tasks)

Lin et al., J. Phys. Condens. Matter **25** (2013), 295501
Lin et al., J. Phys. Condens. Matter **26** (2014), 305503

Dense and sparse matrix representation

Matrices in large systems tend to be sparse

(localized basis functions far from each other do not overlap)

Dense and sparse matrix representation

Matrices in large systems tend to be sparse

(localized basis functions far from each other do not overlap)

dense

2	0	2	0	0
0	1	0	7	0
2	0	0	0	0
0	7	0	9	5
0	0	0	5	0

- ✓ Fast, continuous access to matrix elements
- ✓ Highly optimized linear algebra routines (BLAS) for both CPUs and GPUs
- X $O(N^2)$ memory consumption
- X Not affordable for large N

Dense and sparse matrix representation

Matrices in large systems tend to be sparse

(localized basis functions far from each other do not overlap)

dense

2	0	2	0	0
0	1	0	7	0
2	0	0	0	0
0	7	0	9	5
0	0	0	5	0

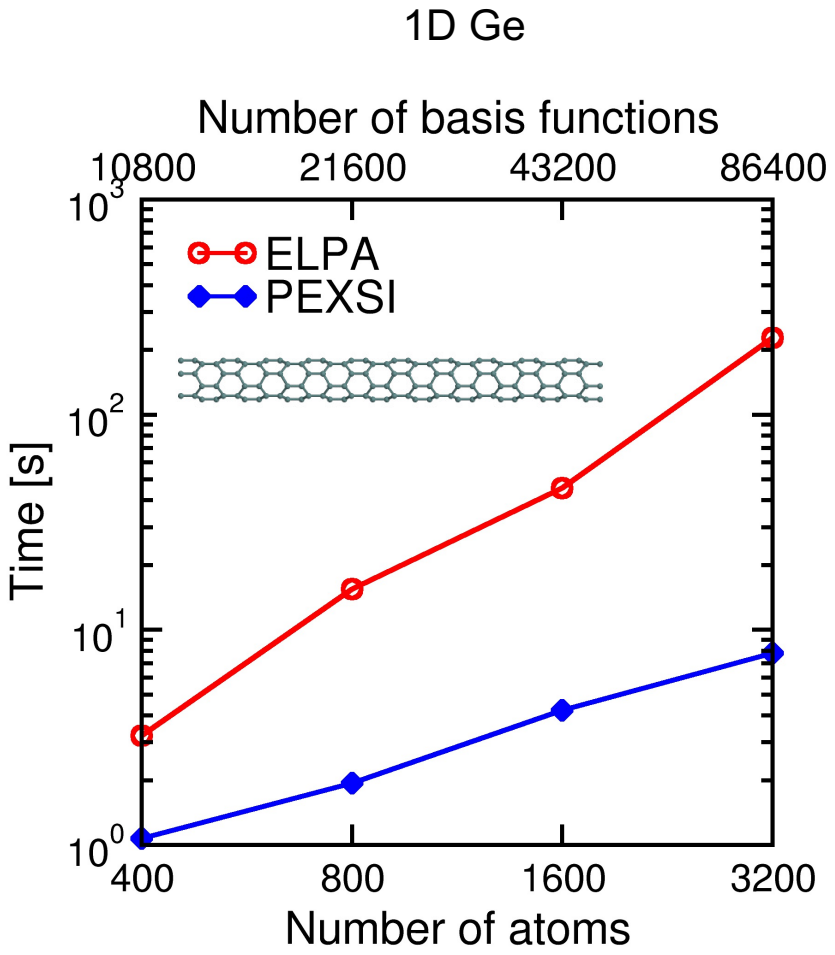
sparse (CSC format)

nnz_val	2	2	1	7	2	7	9	5	5
row_idx	1	3	2	4	1	2	4	5	4
col_ptr	1	3	5	6	9	10			

- ✓ Fast, continuous access to matrix elements
- ✓ Highly optimized linear algebra routines (BLAS) for both CPUs and GPUs
- X $O(N^2)$ memory consumption
- X Not affordable for large N

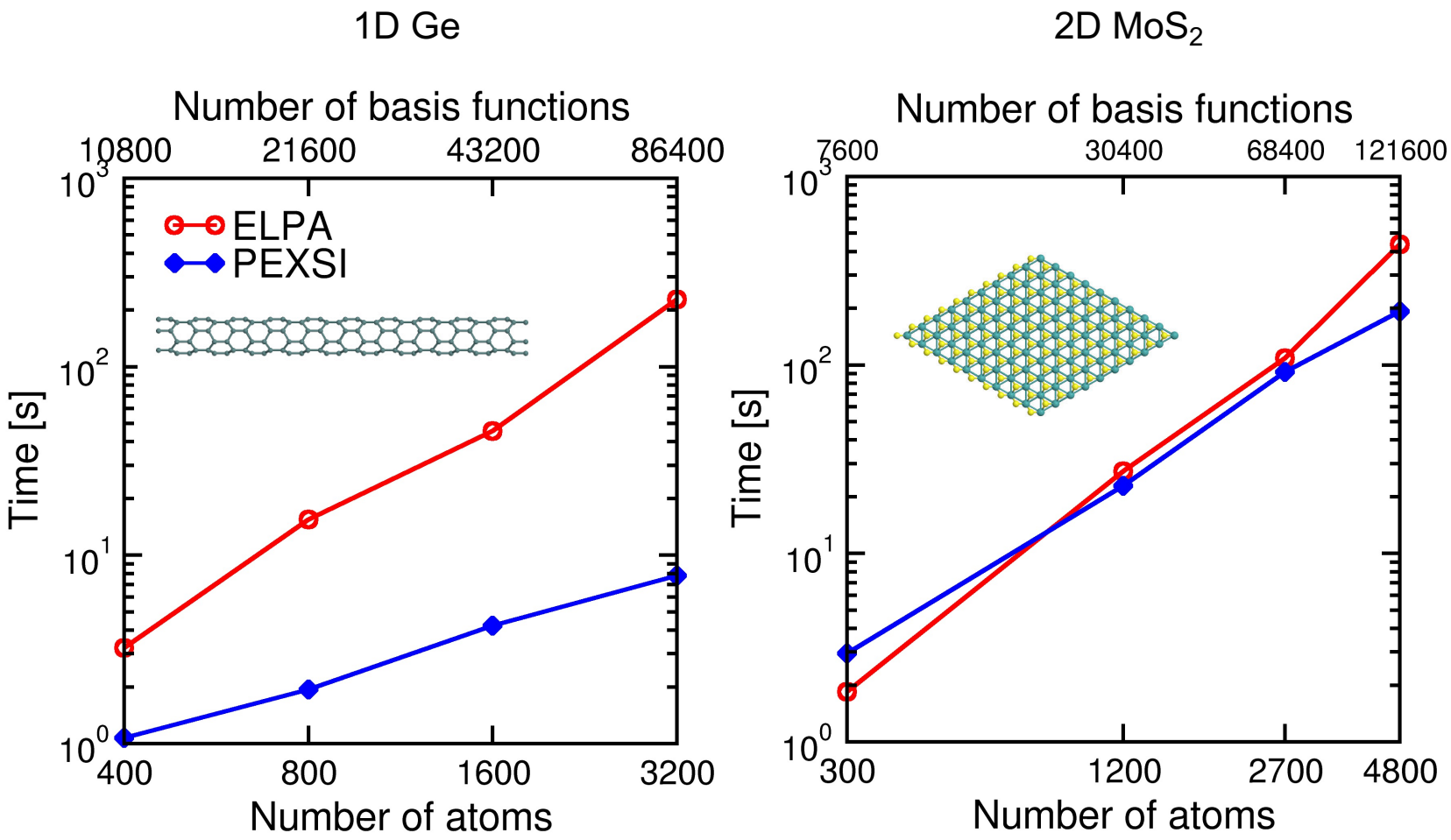
- ✓ O(N) memory consumption
- ✓ Lower operation count (zeros not computed)
- X Indirect access to matrix elements
- ? Customized BLAS routines

Pole expansion and selected inversion (PEXSI)



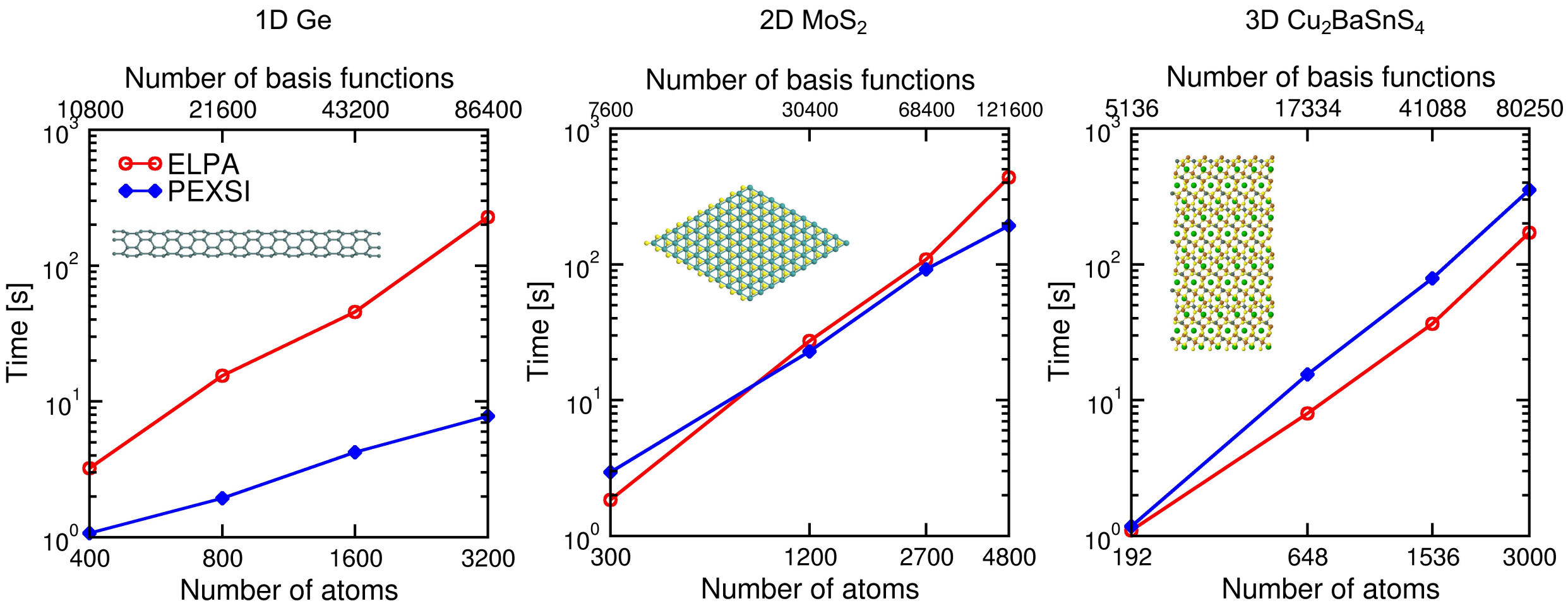
DFT-PBE calculations with the FHI-aims code, 2,560 CPU cores on Cori-Haswell

Pole expansion and selected inversion (PEXSI)



DFT-PBE calculations with the FHI-aims code, 2,560 CPU cores on Cori-Haswell

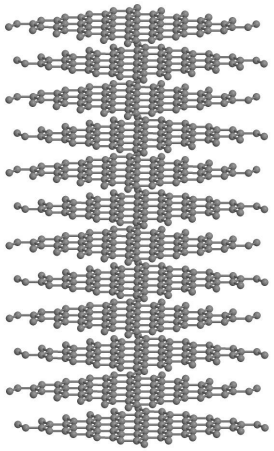
Pole expansion and selected inversion (PEXSI)

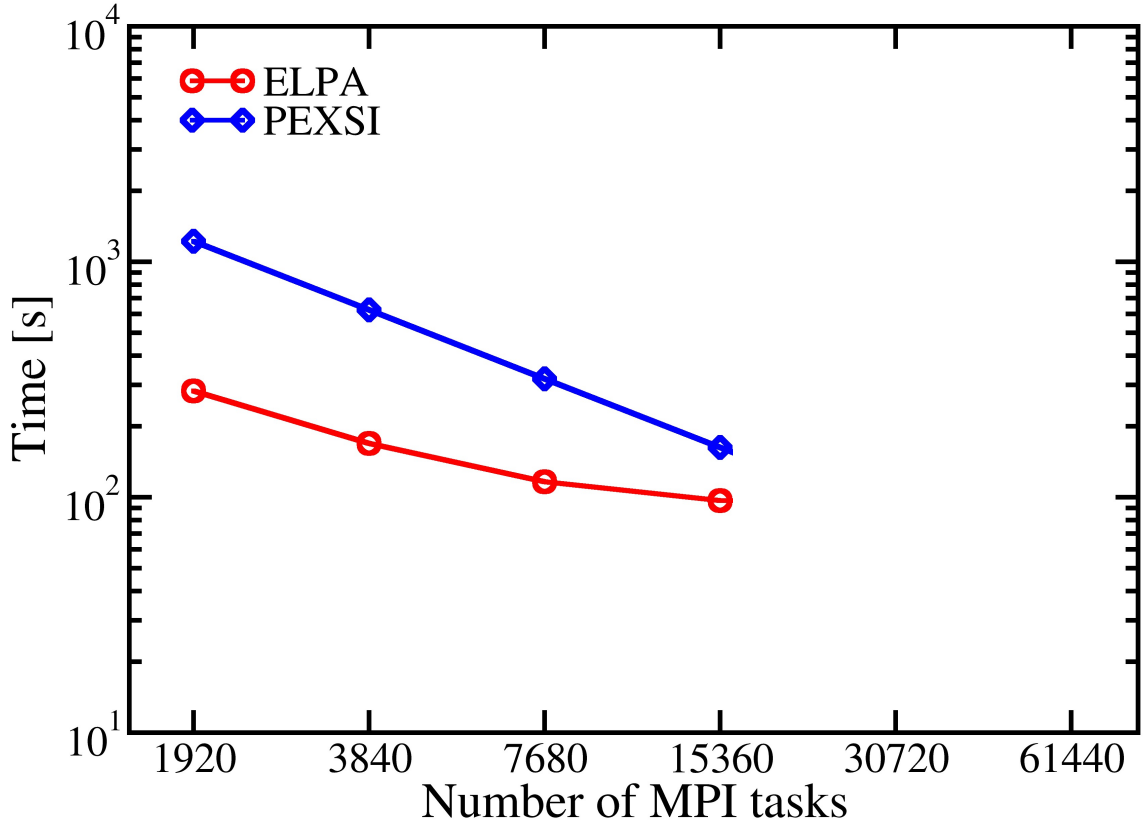


DFT-PBE calculations with the FHI-aims code, 2,560 CPU cores on Cori-Haswell

Pole expansion and selected inversion (PEXSI)

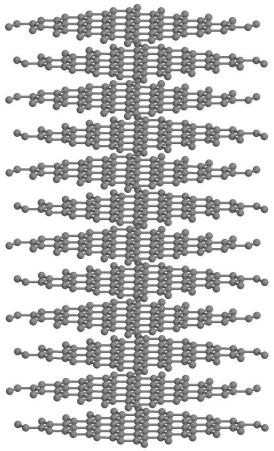
3D graphite
FHI-aims, DFT-PBE
6,912 atoms
96,768 basis functions



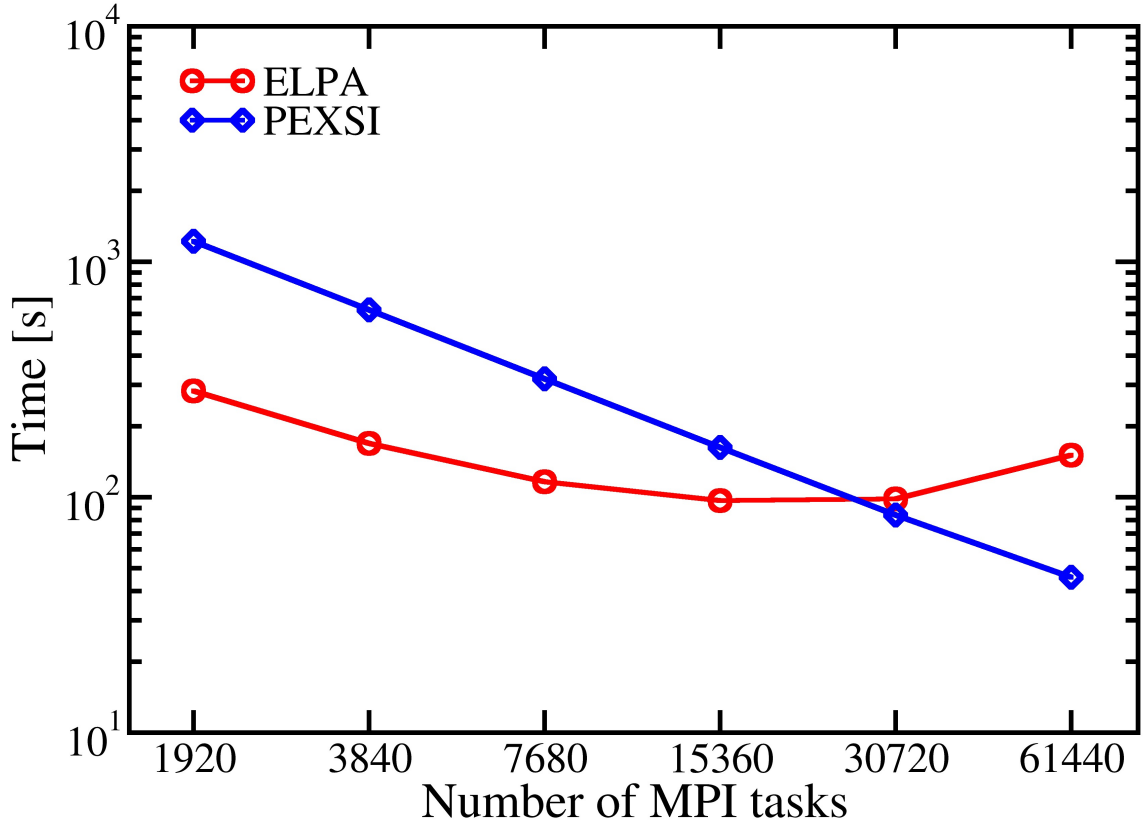


Pole expansion and selected inversion (PEXSI)

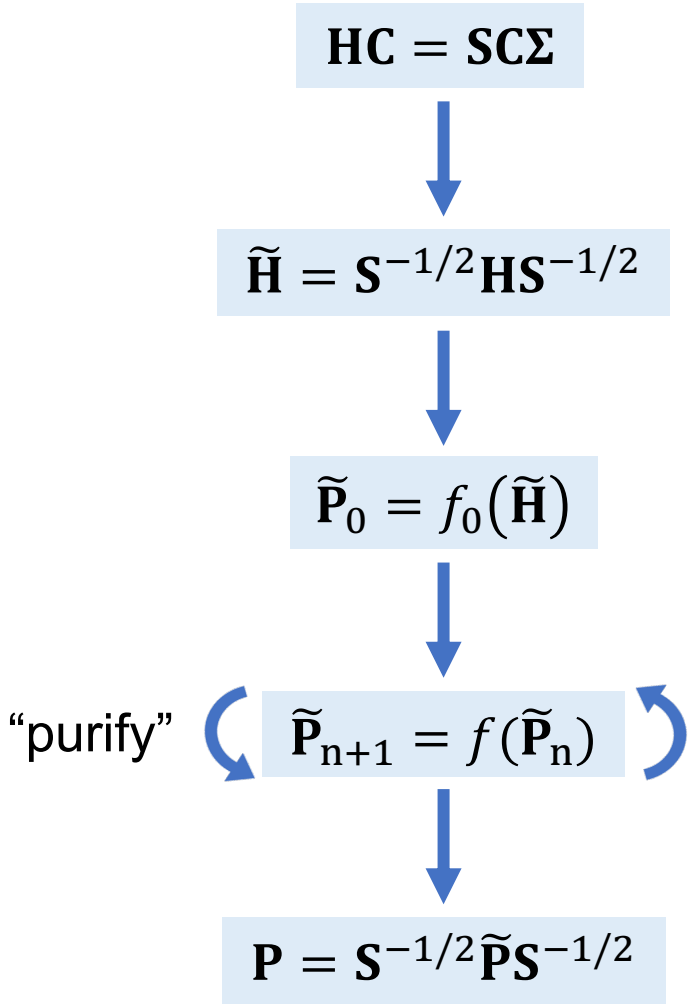
3D graphite
FHI-aims, DFT-PBE
6,912 atoms
96,768 basis functions



ELPA scales to ~20k CPU cores
PEXSI displays almost ideal scaling



Density matrix purification (NTPoly)



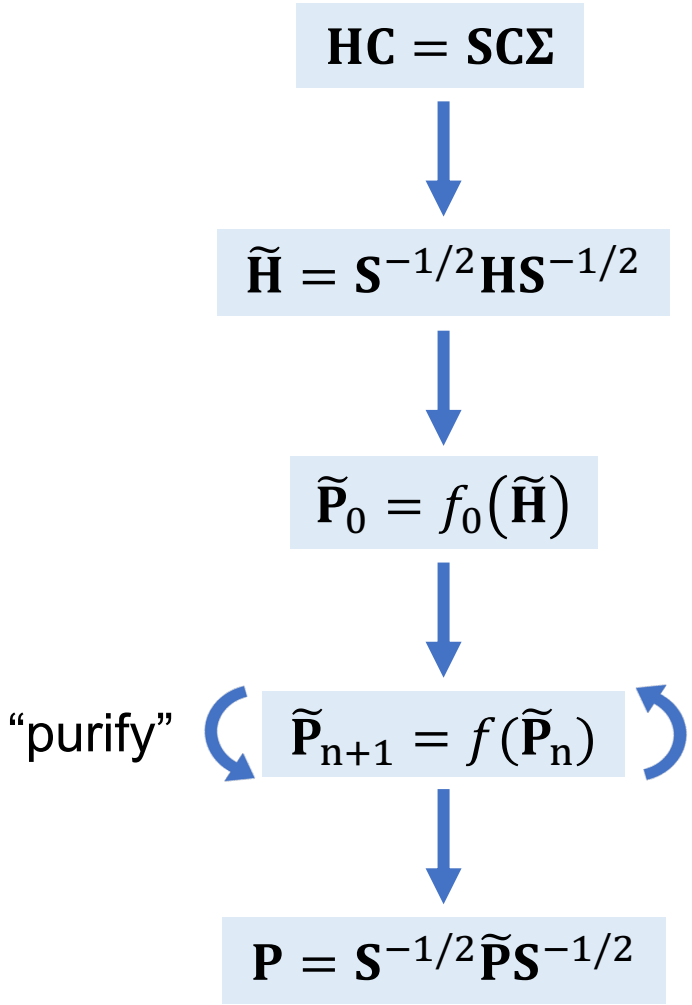
- Iteratively "purify" density matrix to satisfy:
  - Hermitian: $\mathbf{P} = \mathbf{P}^*$
  - Normalized: $\mathrm{Tr}(\mathbf{P}) = \mathrm{N}_{\mathrm{electron}}$
  - Idempotent: $\mathbf{P} = \mathbf{P}^2$
- $f(\widetilde{\mathbf{P}}_n)$: Usually a matrix polynomial
- Example: 2nd order trace resetting purification

$$\sigma_n = \mathrm{sign}\left(\mathrm{N}_{\mathrm{electron}} - \mathrm{Tr}(\mathbf{P}_n)\right)$$
$$\mathbf{P}_{n+1} = \mathbf{P}_n + \sigma_n(\mathrm{I} - \mathbf{P}_n)\mathbf{P}_n = (1 + \sigma_n)\mathbf{P}_n - \sigma_n\mathbf{P}_n^2$$

- Linear scaling via sparse matrix multiplication

Goedecker, Rev. Mod. Phys. **71** (1999), 1085-1123
Bowler and Miyazaki, Rep. Prog. Phys. **75** (2012), 036503
Dawson and Nakajima, Comput. Phys. Commun. **225** (2018), 154-165

Density matrix purification (NTPoly)



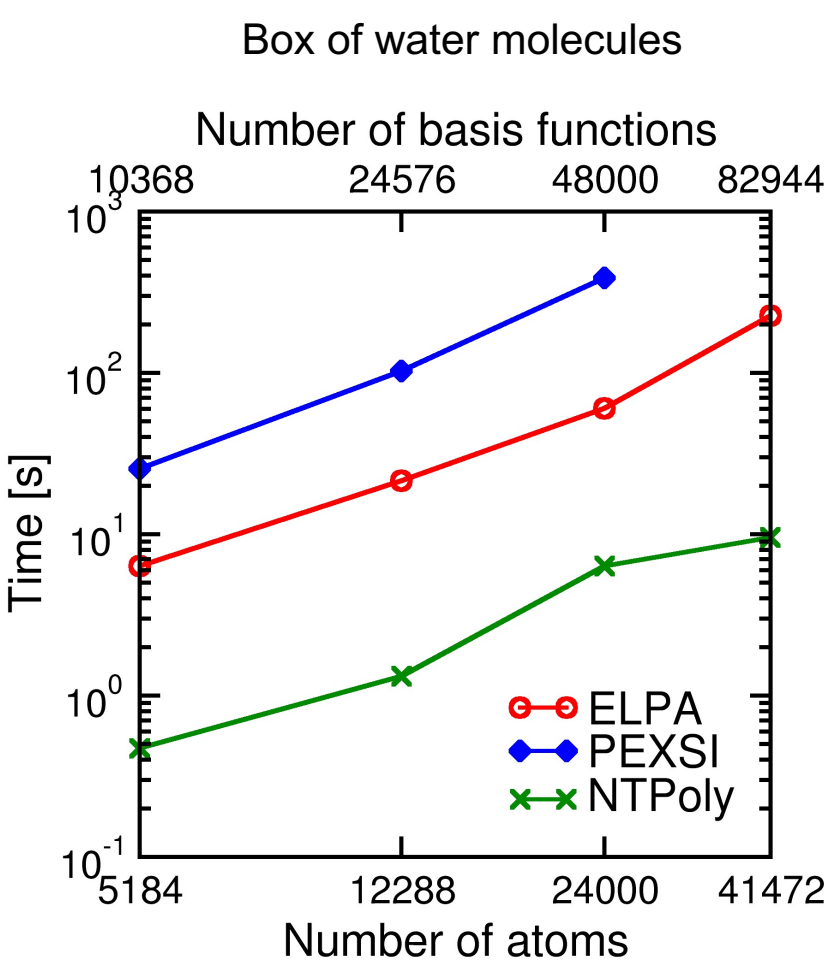
- Iteratively "purify" density matrix to satisfy:
  - Hermitian: $\mathbf{P} = \mathbf{P}^*$
  - Normalized: $\mathrm{Tr}(\mathbf{P}) = \mathrm{N}_{\mathrm{electron}}$
  - Idempotent: $\mathbf{P} = \mathbf{P}^2$
- $f(\widetilde{\mathbf{P}}_n)$: Usually a matrix polynomial
- Example: 2nd order trace resetting purification

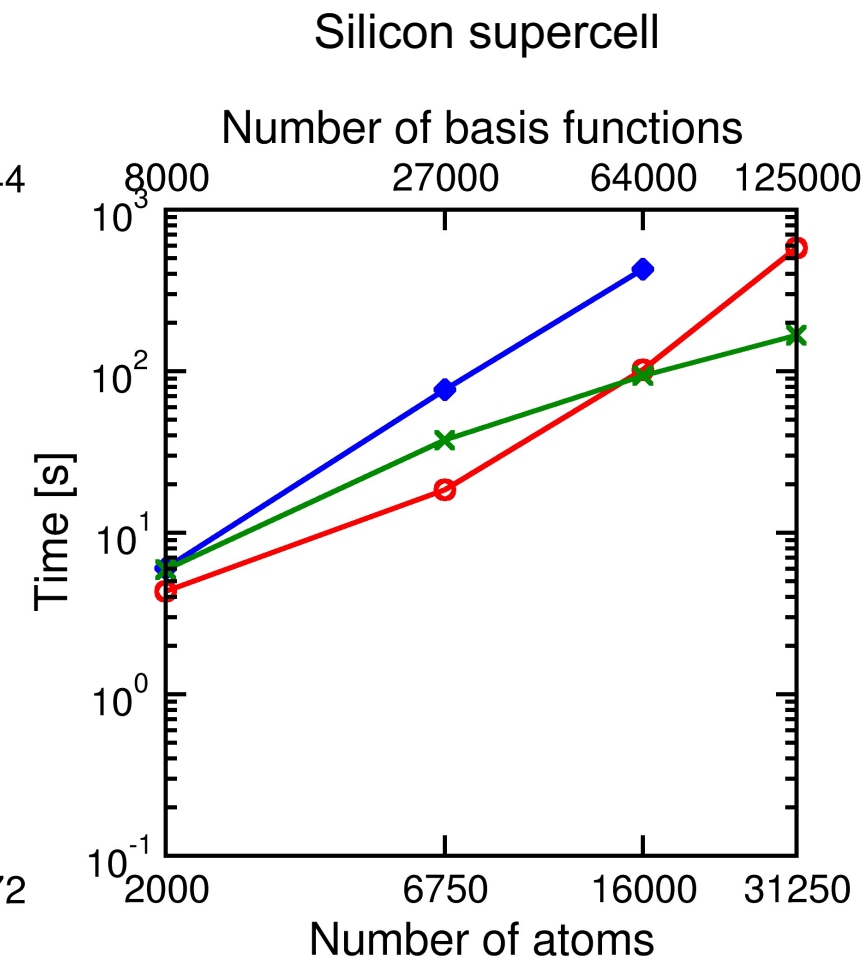
$$\sigma_n = \mathrm{sign}\left(\mathrm{N}_{\mathrm{electron}} - \mathrm{Tr}(\mathbf{P}_n)\right)$$
$$\mathbf{P}_{n+1} = \mathbf{P}_n + \sigma_n(\mathrm{I} - \mathbf{P}_n)\mathbf{P}_n = (1 + \sigma_n)\mathbf{P}_n - \sigma_n\mathbf{P}_n^2$$

- Linear scaling via sparse matrix multiplication
- Linear scaling density-functional perturbation theory (DFPT) in FHI-aims powered by NTPoly

Shang et al., Comput. Phys. Commun. **258** (2021), 107613

Density matrix purification (NTPoly)





NTPoly settings:

- 4^{th} order TRS method
- 10^{-5} truncation threshold
- 10^{-2} convergence criterion

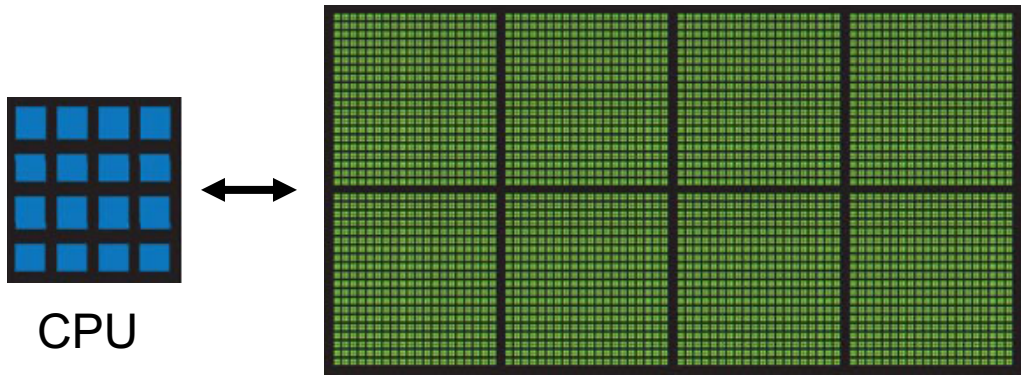
Same accuracy: Band structure energies agree across the three solvers within 10^{-5} eV/atom

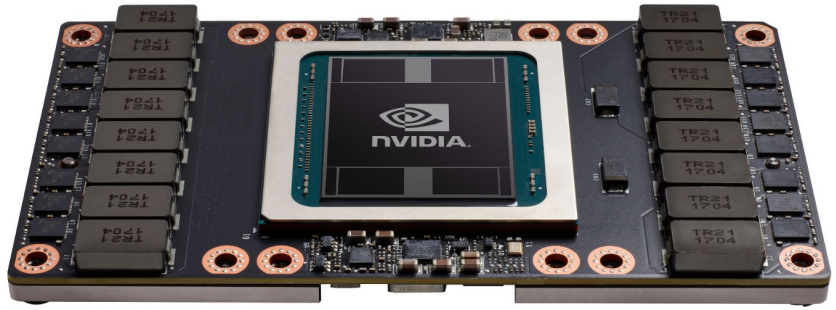
DFTB calculations with the DFTB+ code, 2,560 CPU cores on Cori-Haswell

GPU computing

Graphics processing unit (GPU)

- Strengths of GPUs
  - High parallel performance (thousands of cores)
  - Power efficiency
- As of today, 6 of the top 10 supercomputers in the world have GPU accelerators
- GPUs will power the first exascale (10^{18} floating point operations per second) supercomputers in the US

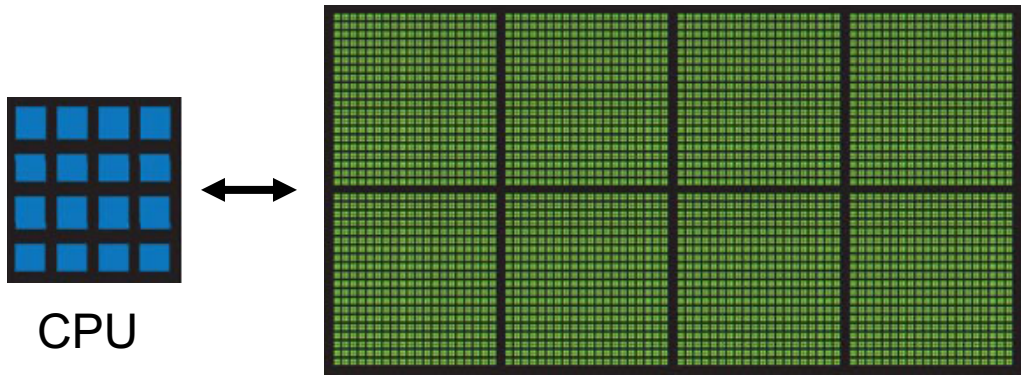


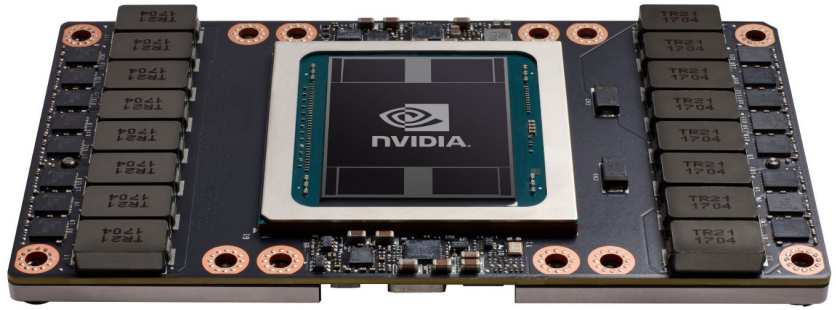


https://developer.nvidia.com/cuda-zone

Graphics processing unit (GPU)

- Strengths of GPUs
  - High parallel performance (thousands of cores)
  - Power efficiency
- As of today, 6 of the top 10 supercomputers in the world have GPU accelerators
- GPUs will power the first exascale (10^{18} floating point operations per second) supercomputers in the US
- GPU acceleration in FHI-aims
  - Hamiltonian integration
  - Electron density
  - Forces and stress tensor
  - Eigensolver (GPU-accelerated ELPA)

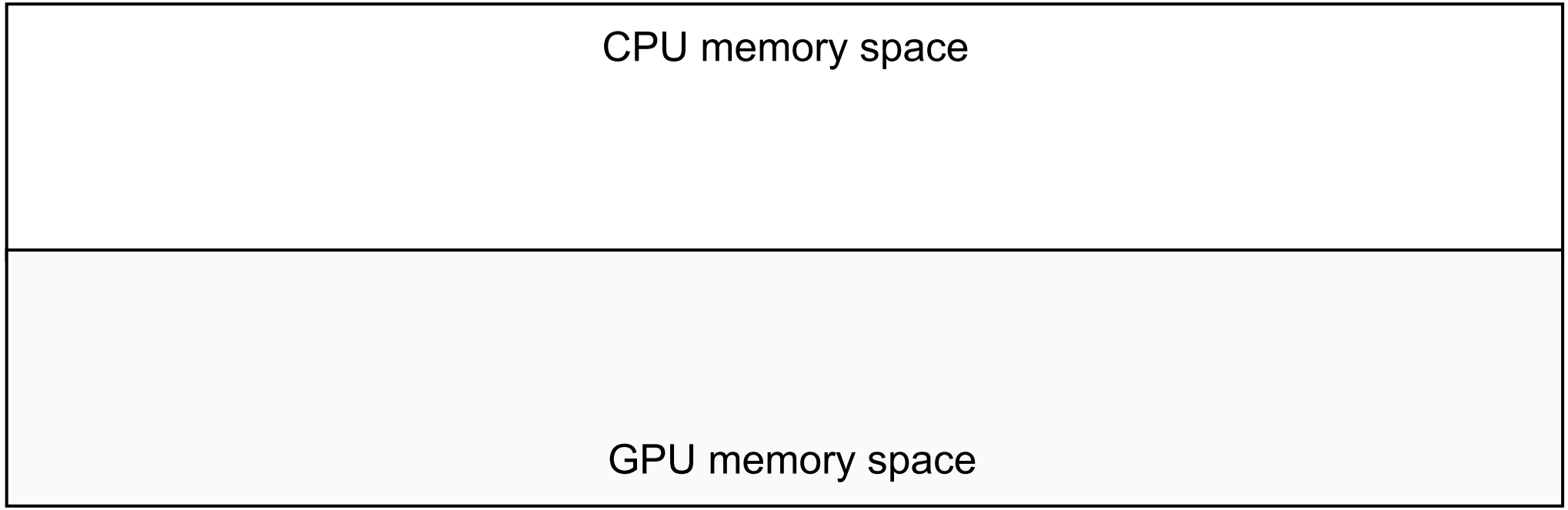




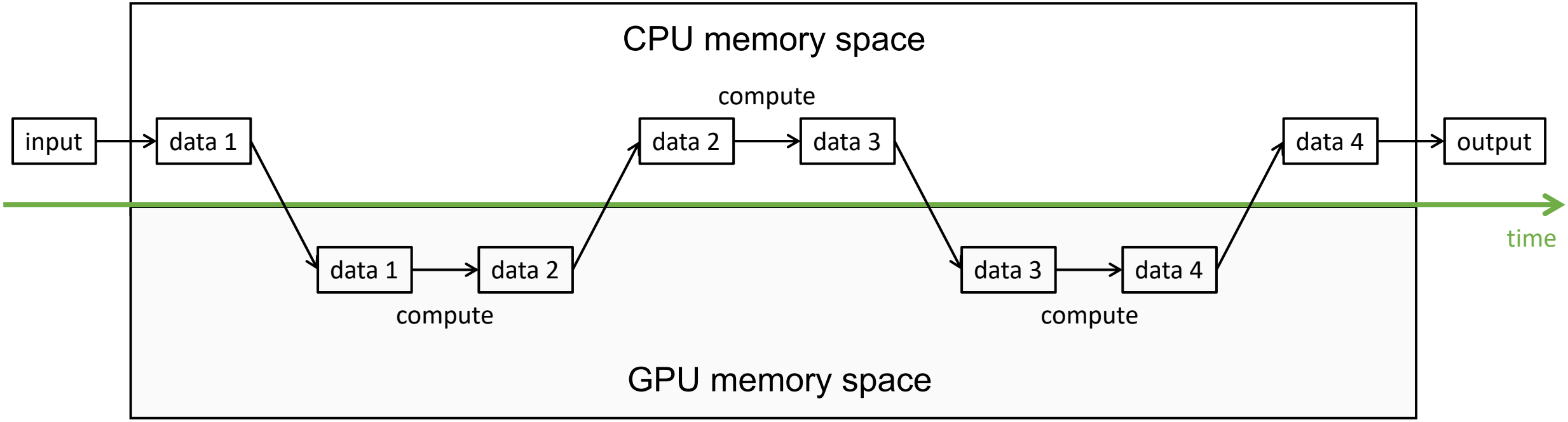
https://developer.nvidia.com/cuda-zone

Huhn et al., Comput. Phys. Commun. **254** (2020), 107314
Yu et al., Comput. Phys. Commun. **262** (2021), 107808

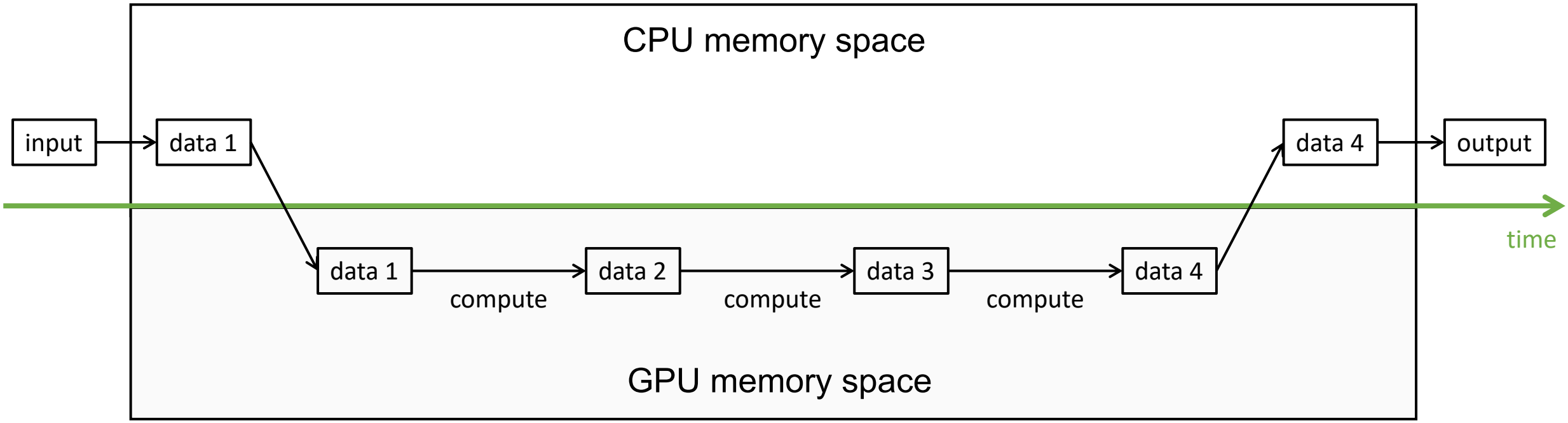
GPU computing



GPU computing

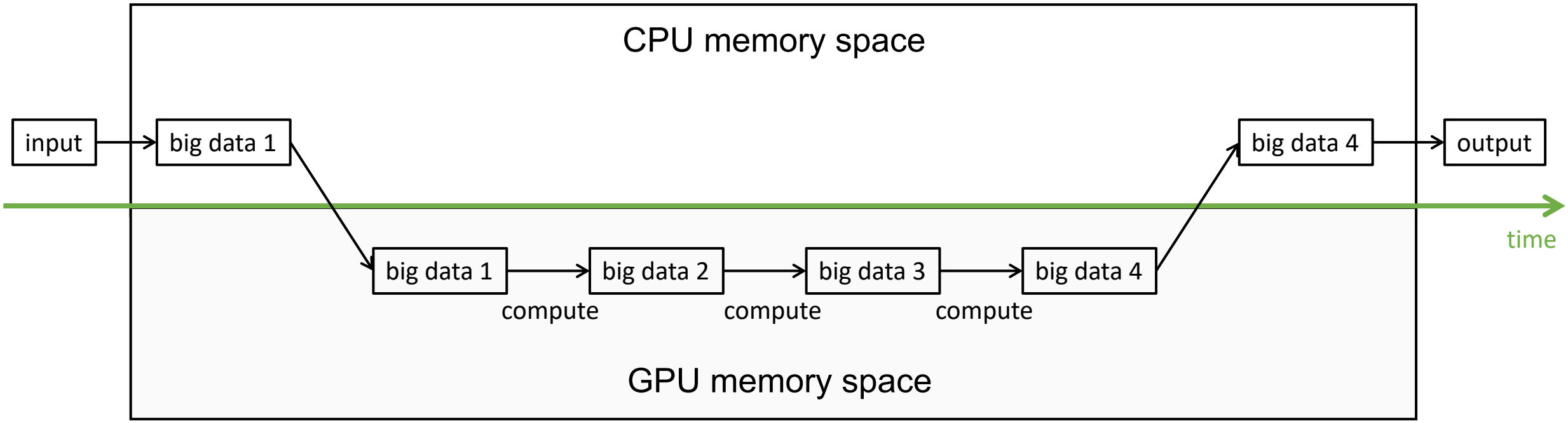


GPU computing



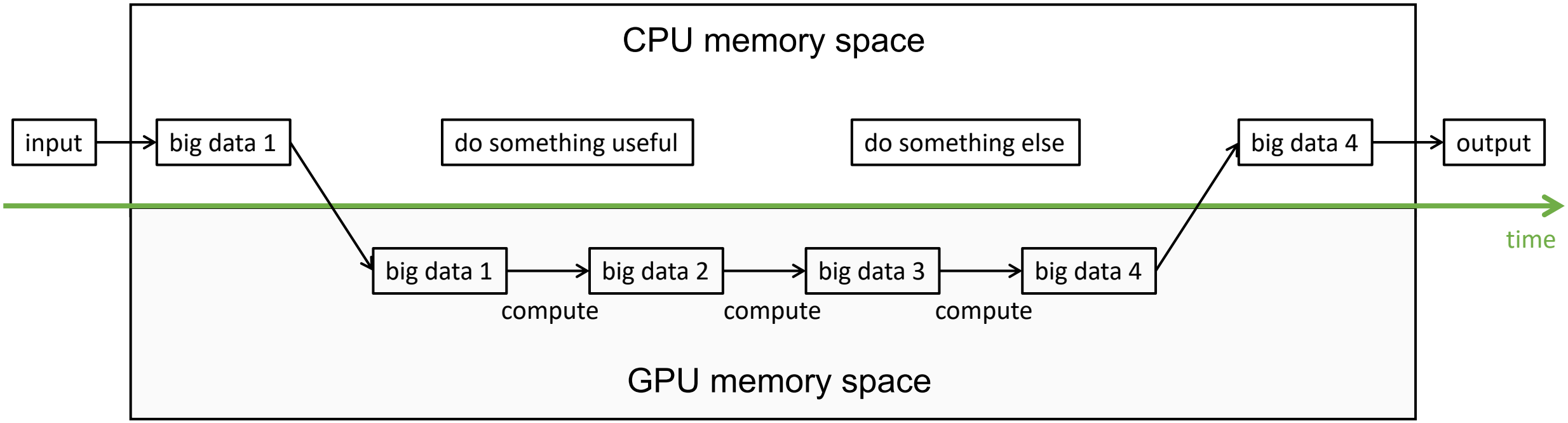
- Reduce communication

GPU computing



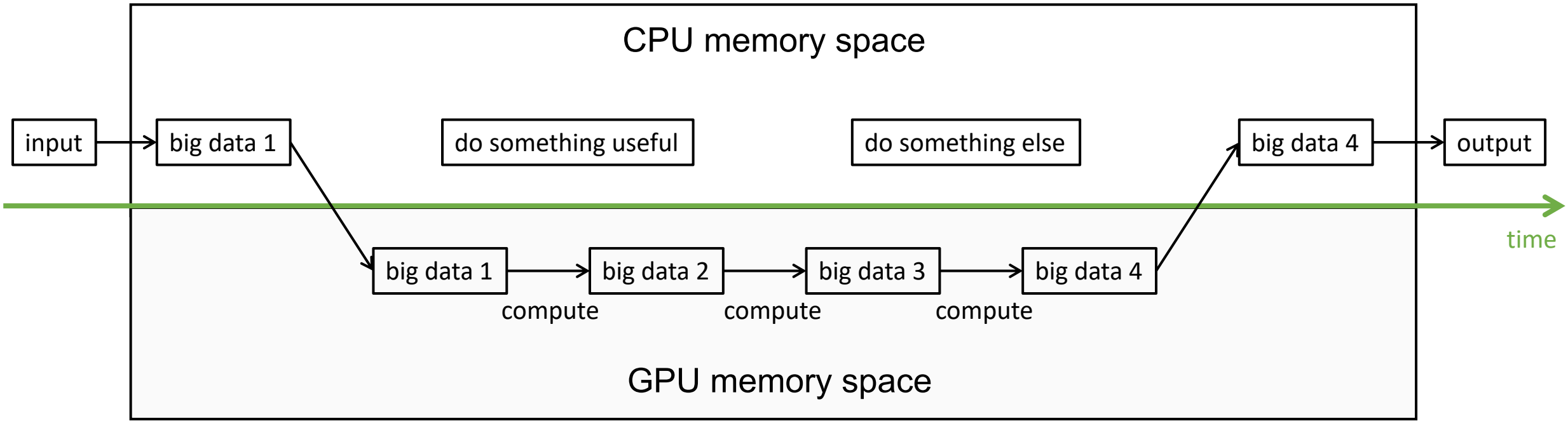
- Reduce communication
- Keep GPU saturated

GPU computing



- Reduce communication
- Keep GPU saturated
- Overlap CPU and GPU activities

GPU computing

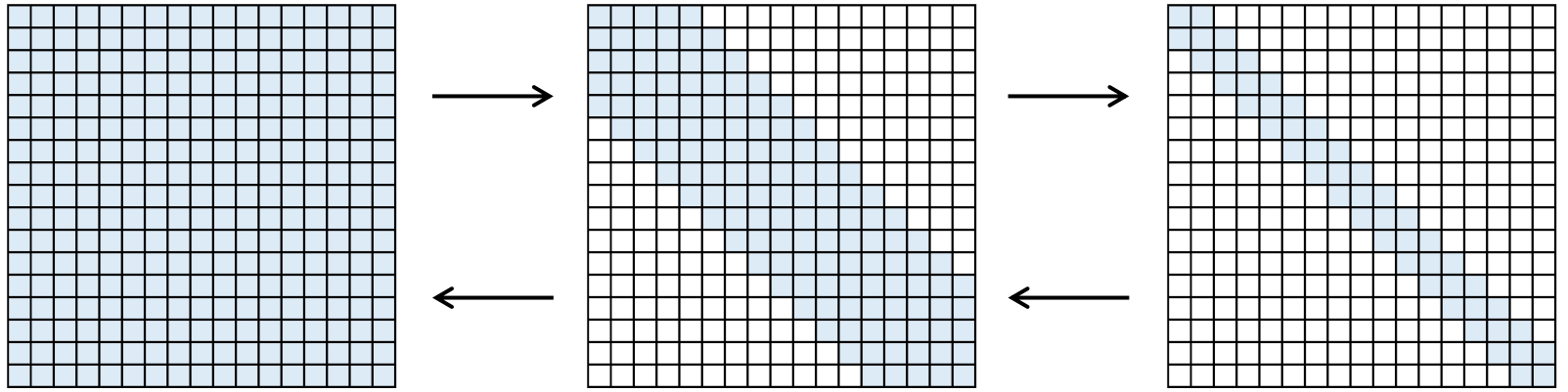


- Reduce communication
- Keep GPU saturated
- Overlap CPU and GPU activities

GPU-accelerated ELPA eigensolver

Yu et al., Comput. Phys. Commun. **262** (2021), 107808

GPU acceleration in the ELPA eigensolver



Two-stage diagonalization

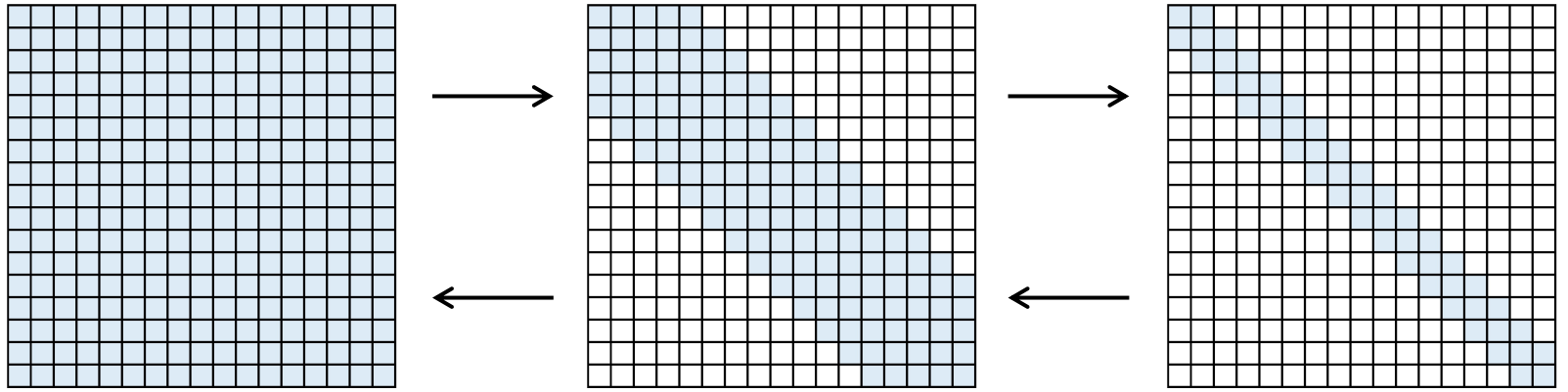
1) Full matrix A → banded matrix B
2) Banded matrix B → tridiagonal matrix T
3) Solve tridiagonal T
4) Tridiagonal eigenvectors → banded
5) Banded eigenvectors → full

GPU-accelerated dense linear algebra by cuBLAS

GPU-accelerated Householder transformations by CUDA

Minimized communication between CPUs and GPUs

GPU acceleration in the ELPA eigensolver



Two-stage diagonalization

1) Full matrix A → banded matrix B
2) Banded matrix B → tridiagonal matrix T
3) Solve tridiagonal T
4) Tridiagonal eigenvectors → banded
5) Banded eigenvectors → full

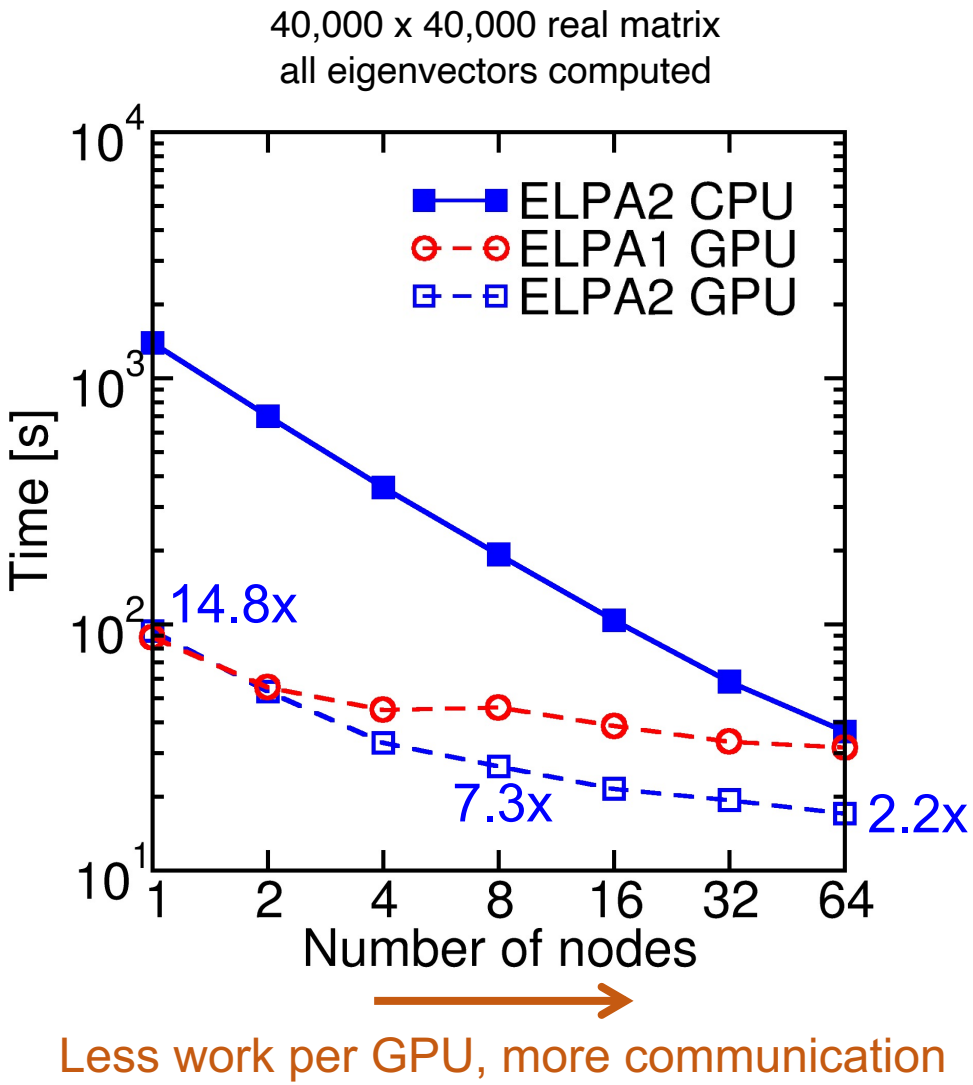
GPU-accelerated dense linear algebra by cuBLAS
GPU-accelerated Householder transformations by CUDA
Minimized communication between CPUs and GPUs

Largest problem solved by ELPA-GPU:

1,769,472 x 1,769,472 complex matrix
all eigenvalues and eigenvectors
4,000 Summit nodes (24,000 GPUs)

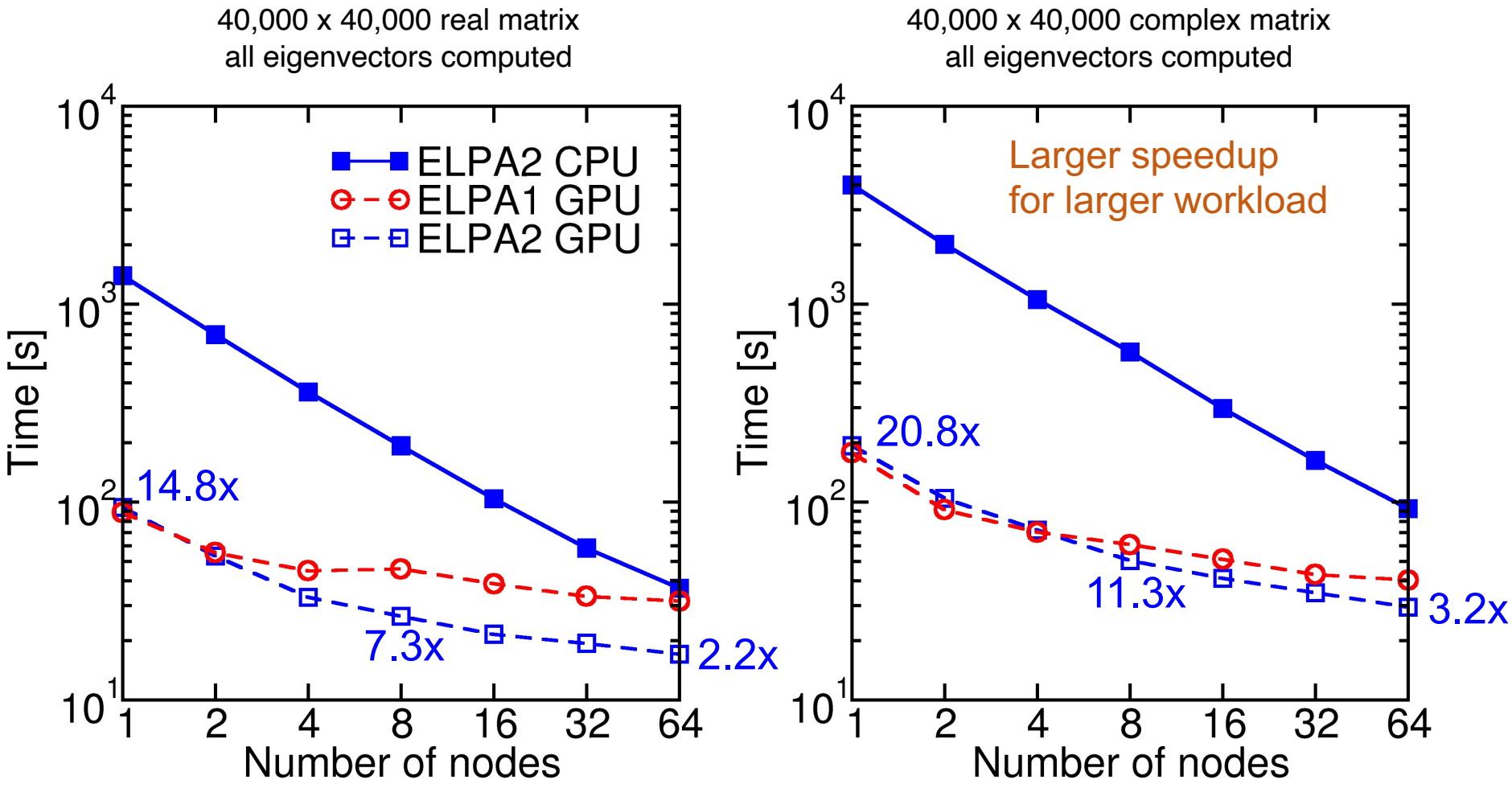
Dr. Gabriel Wlazłowski, Warsaw University of Technology
https://gitlab.fizyka.pw.edu.pl/wtools/wslda

ELPA GPU performance on the Summit supercomputer



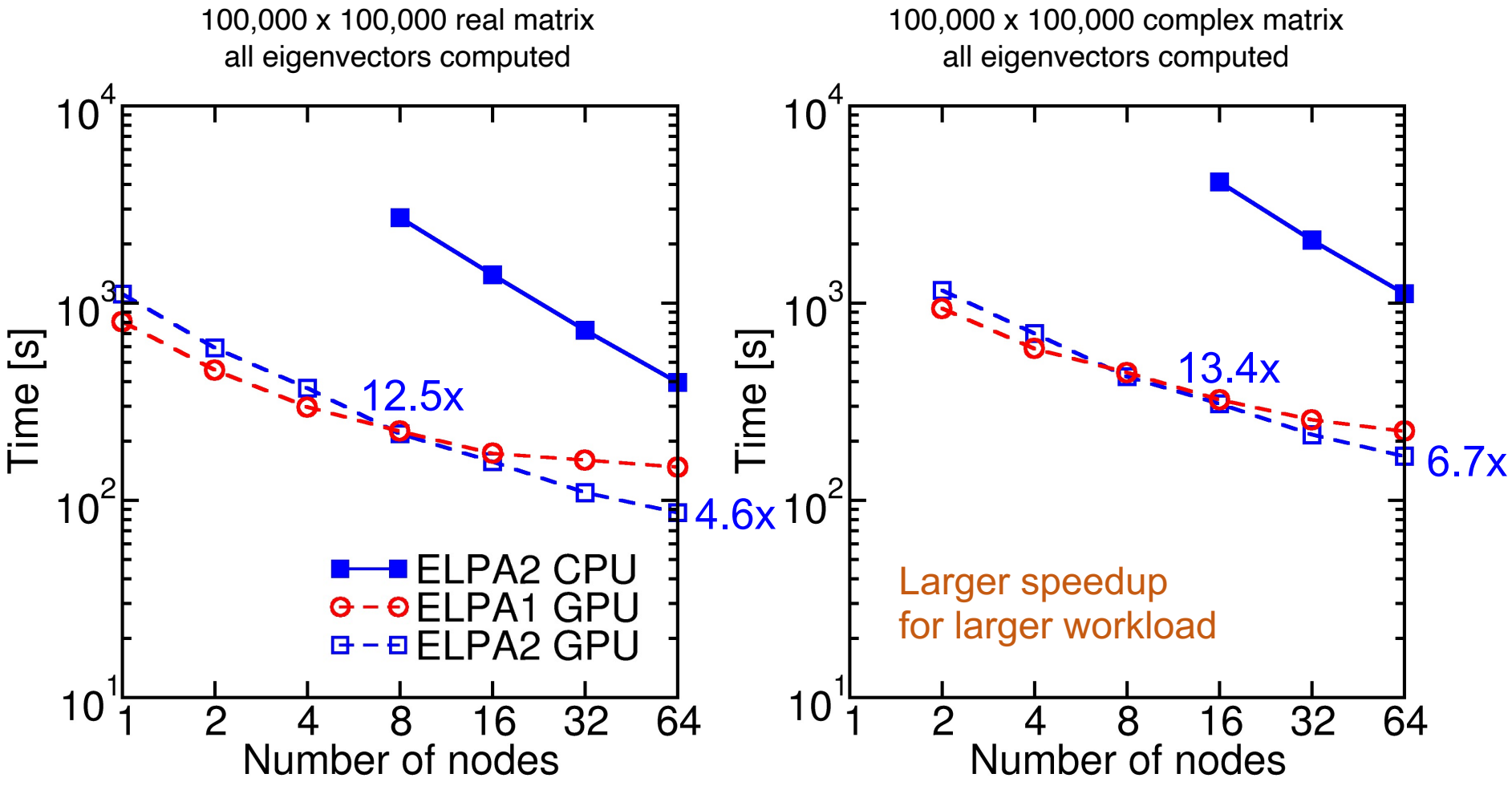
1 Summit node = 42 CPU cores + 6 GPUs

ELPA GPU performance on the Summit supercomputer



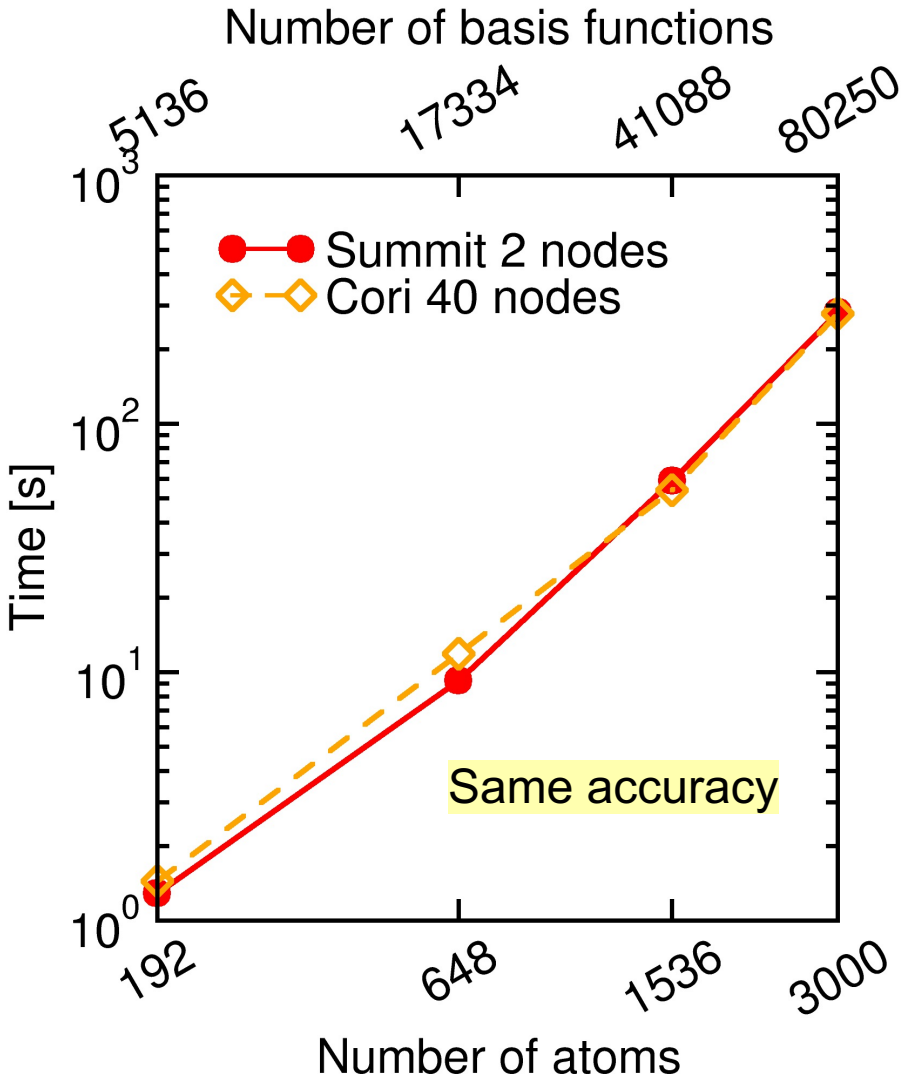
1 Summit node = 42 CPU cores + 6 GPUs

ELPA GPU performance on the Summit supercomputer

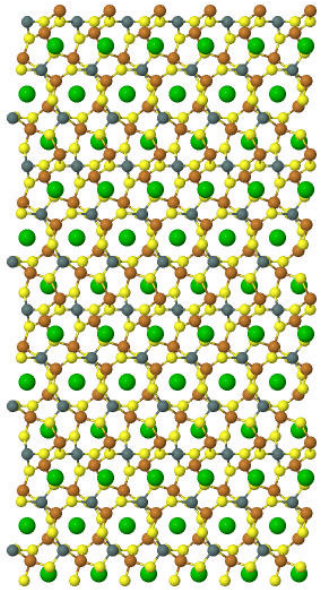


1 Summit node = 42 CPU cores + 6 GPUs

ELPA GPU in FHI-aims



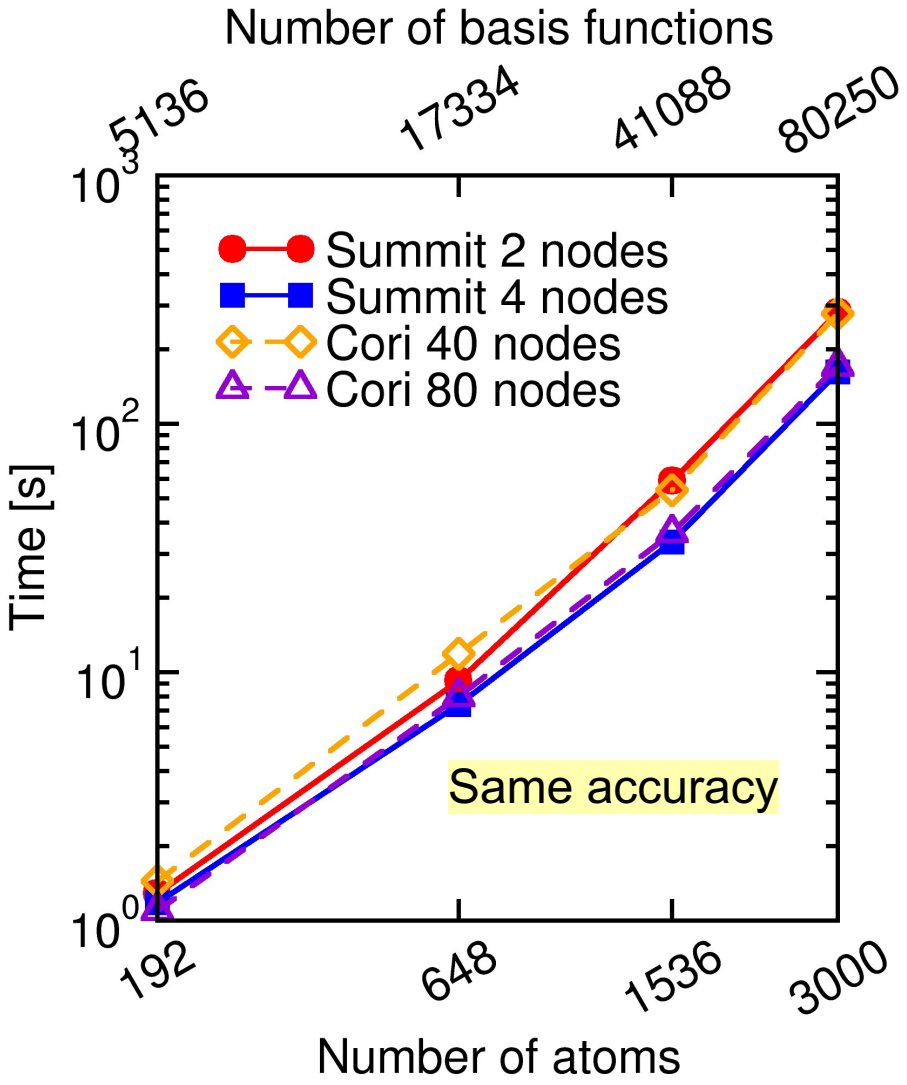
Cu_2BaSnS_4



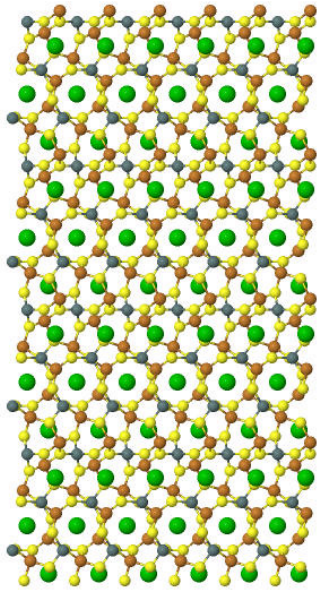
FHI-aims DFT-PBE

- 1 Summit node:
  - 42 IBM POWER9 CPU cores
  - 6 NVIDIA V100 GPUs
- 1 Cori node:
  - 32 Intel Haswell CPU cores

ELPA GPU in FHI-aims



Cu_2BaSnS_4



FHI-aims DFT-PBE

- 1 Summit node:
  - 42 IBM POWER9 CPU cores
  - 6 NVIDIA V100 GPUs
- 1 Cori node:
  - 32 Intel Haswell CPU cores

Frozen core approximation

Frozen core approximation

Kohn-Sham eigenvalue problem

$$\mathbf{HC} = \mathbf{SC\Sigma}$$

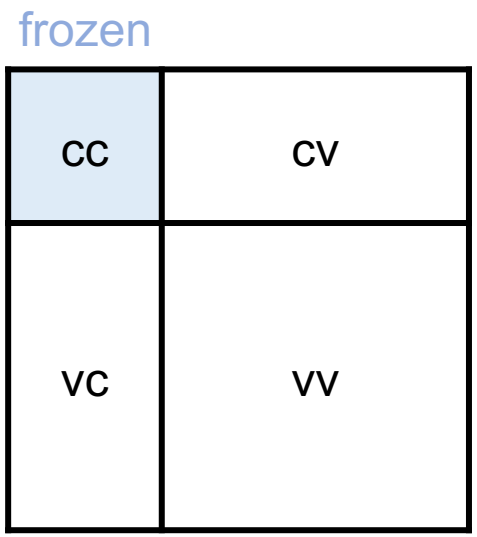
Partition $\mathbf{H}$, $\mathbf{S}$, $\mathbf{C}$ matrices into four blocks

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{cc} & \mathbf{H}_{cv} \\ \mathbf{H}_{vc} & \mathbf{H}_{vv} \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} \mathbf{S}_{cc} & \mathbf{S}_{cv} \\ \mathbf{S}_{vc} & \mathbf{S}_{vv} \end{bmatrix}$$

“Freeze” the core-core (cc) block
→ frozen core Hamiltonian $\hat{\mathbf{H}}_{vv}$

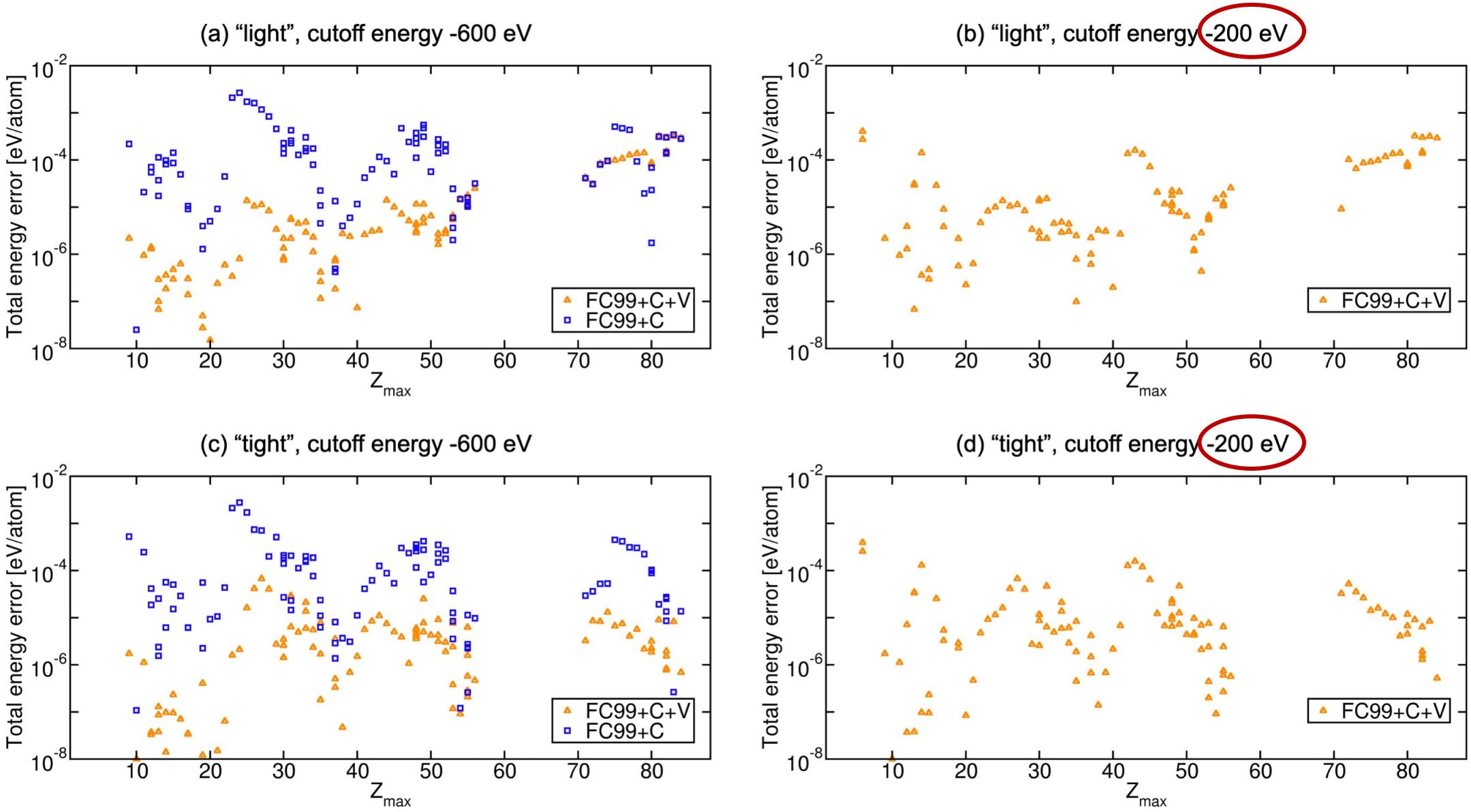
$$\hat{\mathbf{H}}_{vv} = \mathbf{L}_{vv}^{-1}(\mathbf{H}_{vv} + \mathbf{S}_{vc}\mathbf{H}_{cc}\mathbf{S}_{cv} - \mathbf{H}_{vc}\mathbf{S}_{cv} - \mathbf{S}_{vc}\mathbf{H}_{cv})(\mathbf{L}_{vv}^{*})^{-1}$$
$$\mathbf{L}_{vv}\mathbf{L}_{vv}^{*} = \mathbf{S}_{vv} - \mathbf{S}_{vc}\mathbf{S}_{cv}$$

Diagonalize $\hat{\mathbf{H}}_{vv}$ (smaller than $\mathbf{H}$)



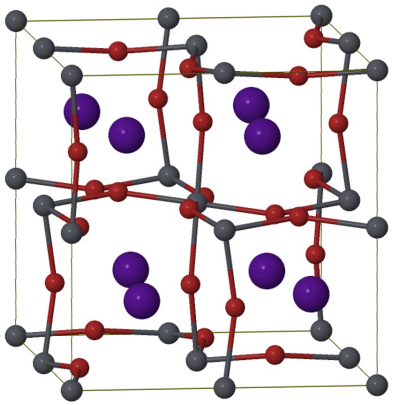
Yu et al., J. Chem. Phys. **154** (2021): 224107

Frozen core approximation



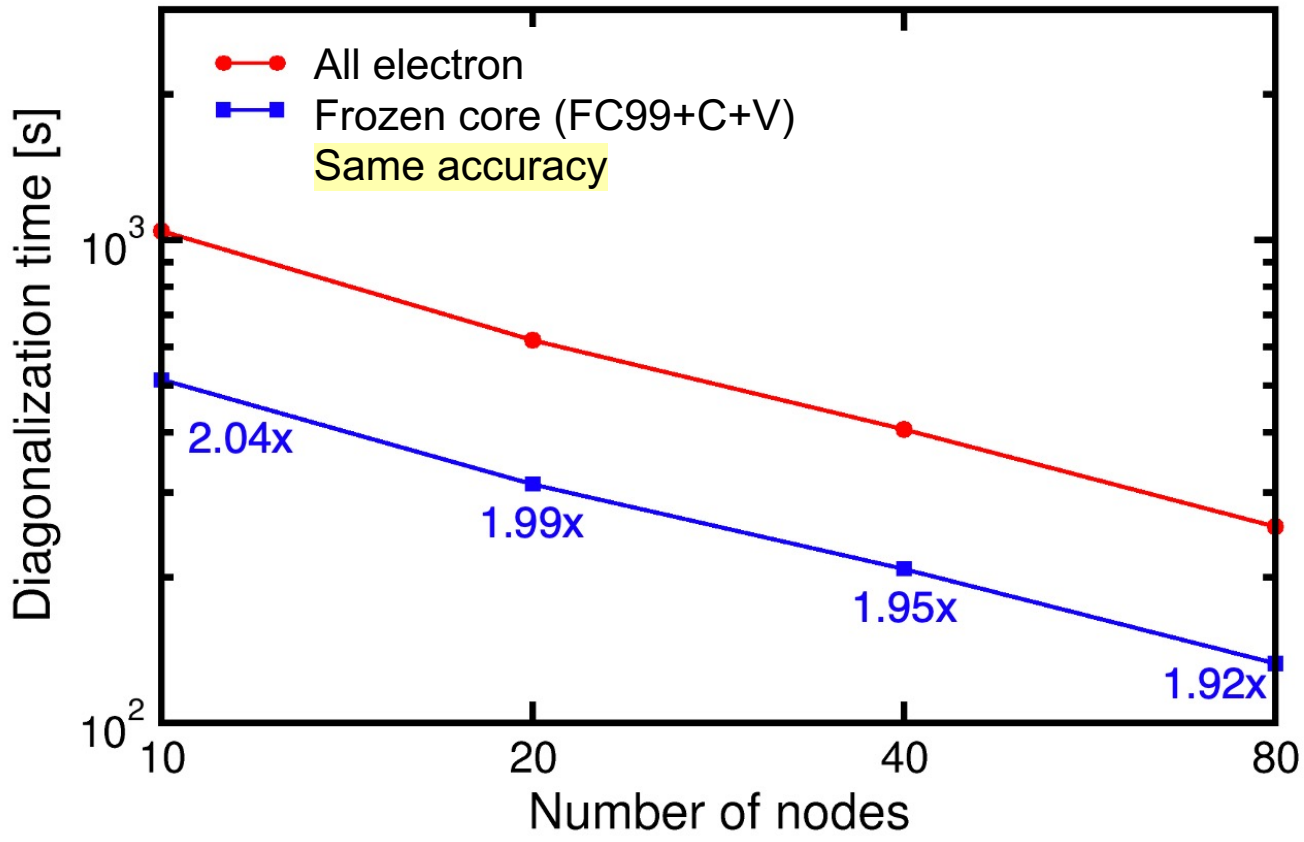
103 compound benchmark set: Huhn and Blum, Phys. Rev. Materials **1** (2017), 033803

Frozen core approximation



FHI-aims, DFT-PBE

$CsPbBr_3$
4x4x4 supercell
2,560 atoms
94,208 basis functions
28,672 frozen
Cori supercomputer (Haswell partition)



Summary

Large KS-DFT simulations can be done with FHI-aims

- Linear scaling integration scheme (previous talk by Dr. Ville Havu)
- Many scalable electronic structure solvers through the ELSI interface
  - ELPA2 eigensolver highly efficient for systems with up to 1k atoms
  - Consider density matrix solvers for sparse, low-dimensional systems
- Leverage the power of GPUs
- Frozen core approximation useful for heavy elements

Happy computing !!

Summary

Large KS-DFT simulations can be done with FHI-aims

- Linear scaling integration scheme (previous talk by Dr. Ville Havu)
- Many scalable electronic structure solvers through the ELSI interface
  - ELPA2 eigensolver highly efficient for systems with up to 1k atoms
  - Consider density matrix solvers for sparse, low-dimensional systems
- Leverage the power of GPUs
- Frozen core approximation useful for heavy elements

Happy computing !!

Some exciting achievements in the community

- Interatomic potential from deep learning
- Single/half precision with no loss of accuracy
- Utilization of tens of thousands of GPUs

→ Simulation of more than a hundred million atoms

Das et al., in Proceedings of SC19', IEEE, 2019, pp. 1-11
Jia et al., in Proceedings of SC20', IEEE, 2020, pp. 1-14
Schade et al., arXiv:2104.08245 (2021)

Key references

Blum et al., Comput. Phys. Commun. **180** (2009): 2175-2196
Marek et al., J. Phys. Condens. Matter **26** (2014): 213201
Yu et al., Comput. Phys. Commun. **222** (2018): 267-285
Yu et al., Comput. Phys. Commun. **256** (2020): 107459
Huhn et al., Comput. Phys. Commun. **254** (2020): 107314
Yu et al., Comput. Phys. Commun. **262** (2021): 107808
Yu et al., J. Chem. Phys. **154** (2021): 224107

Acknowledgments