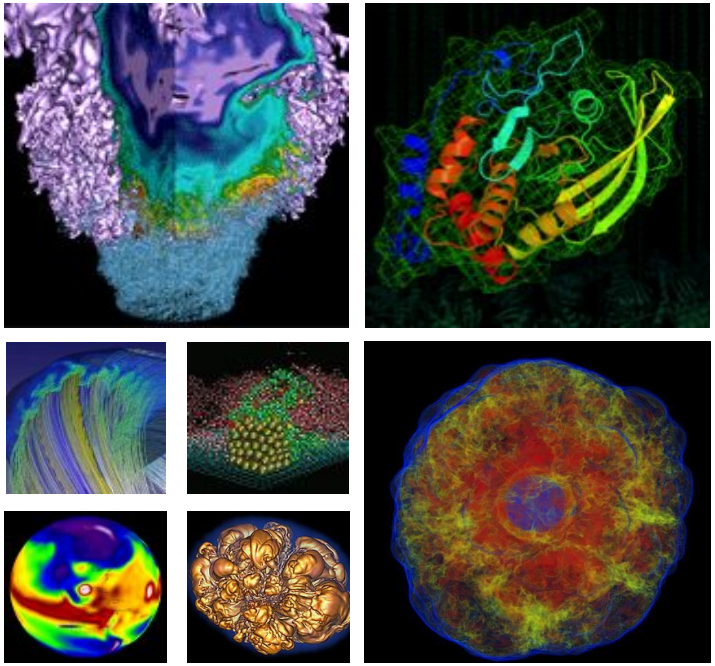
Eigensolver Performance Comparison on Cray XC Systems



NERSC

Brandon Cook, Thorsten Kurth, Jack Deslippe

CRAY

Pierre Carrier, Nick Hill, Nathan Wichmann

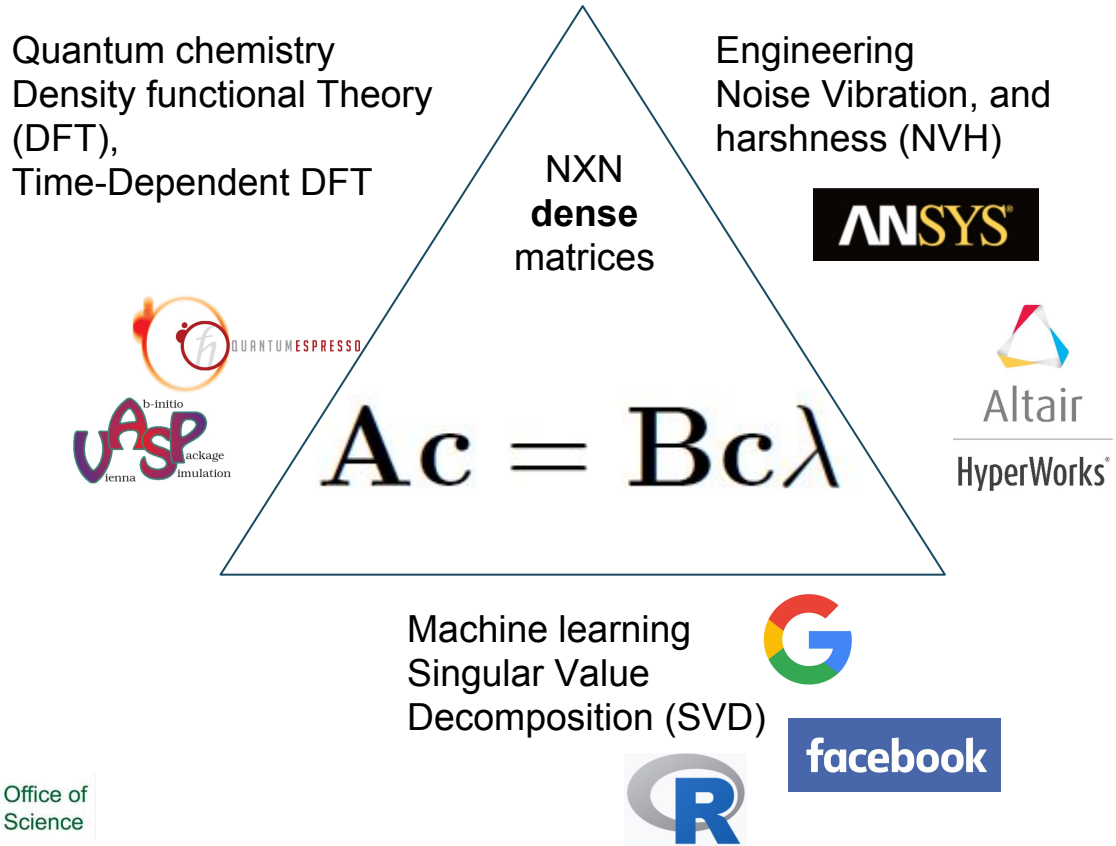
U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

Goal

- Evaluate the general performance of two important eigenvalue solvers: ELPA and SCALAPACK
- Determine rules for optimal parameters
  - Automate the parameter search
- Evaluate OpenMP
- Compare Hardware (Xeon, Xeon-Phi)
- Offer a benchmarking suite (to be continued…)

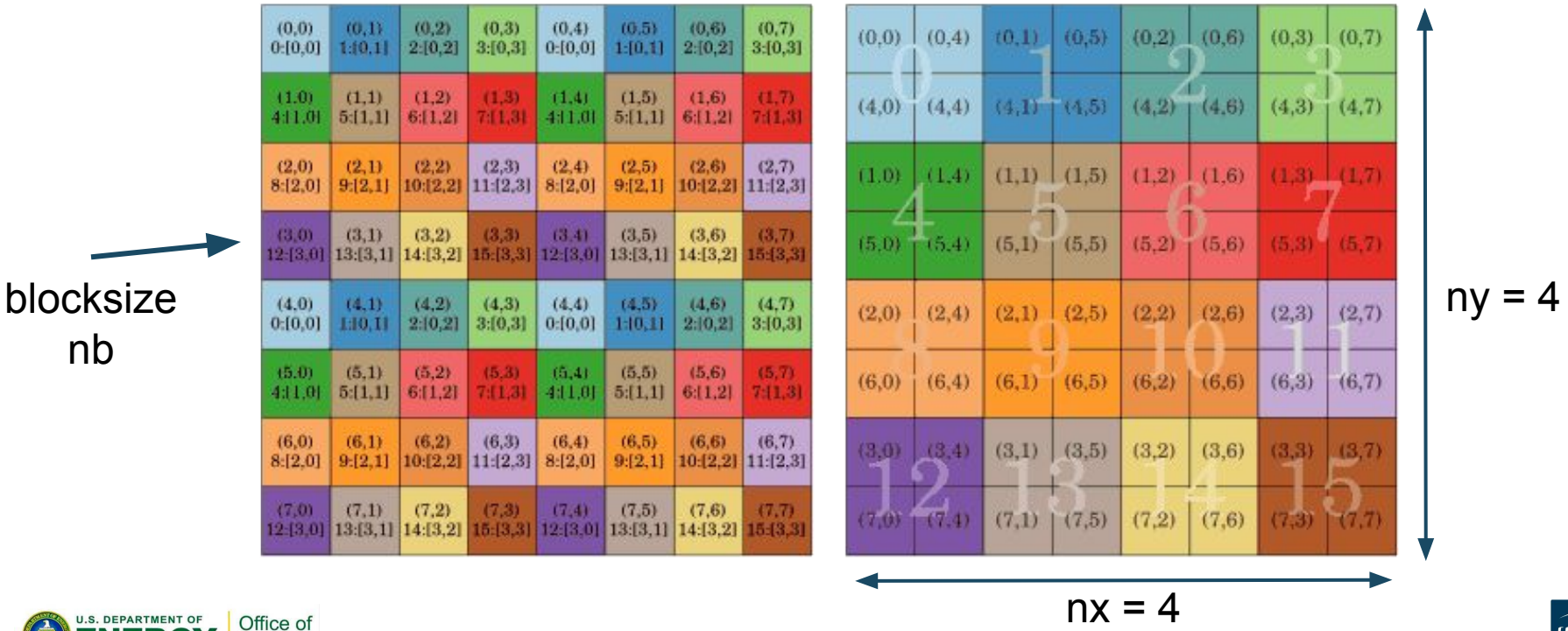
ELPA and ScaLAPACK (PZHEEVD)



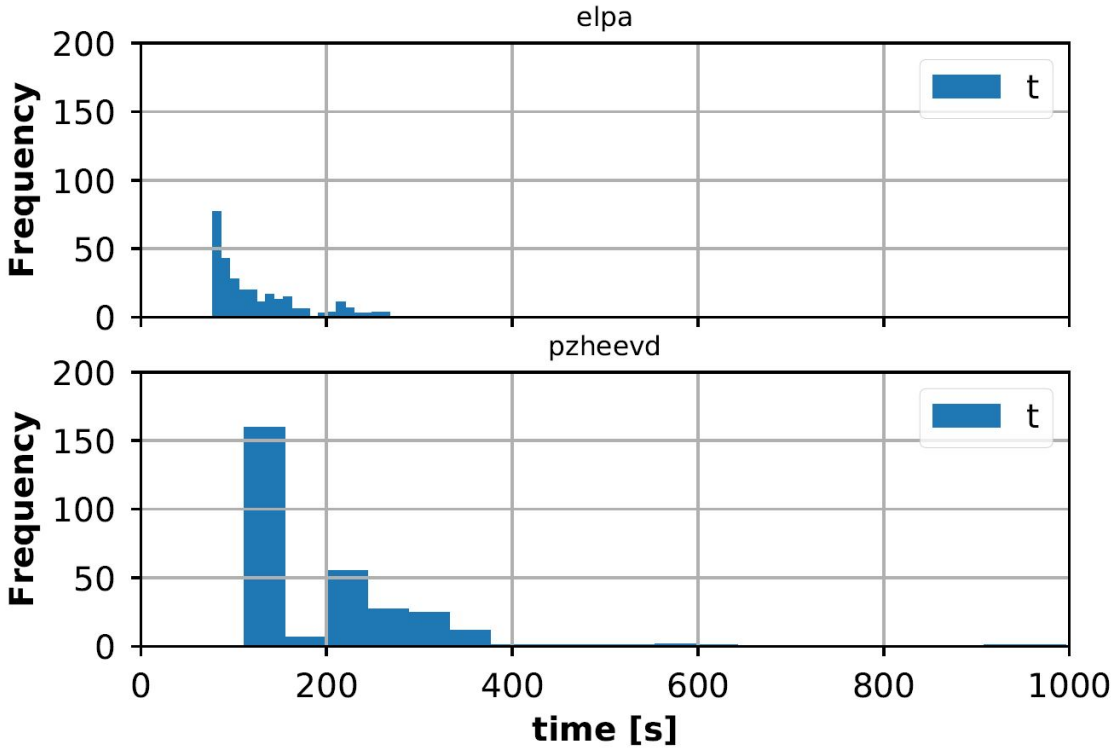
Block cyclic distribution

Global view of array

Process view of array



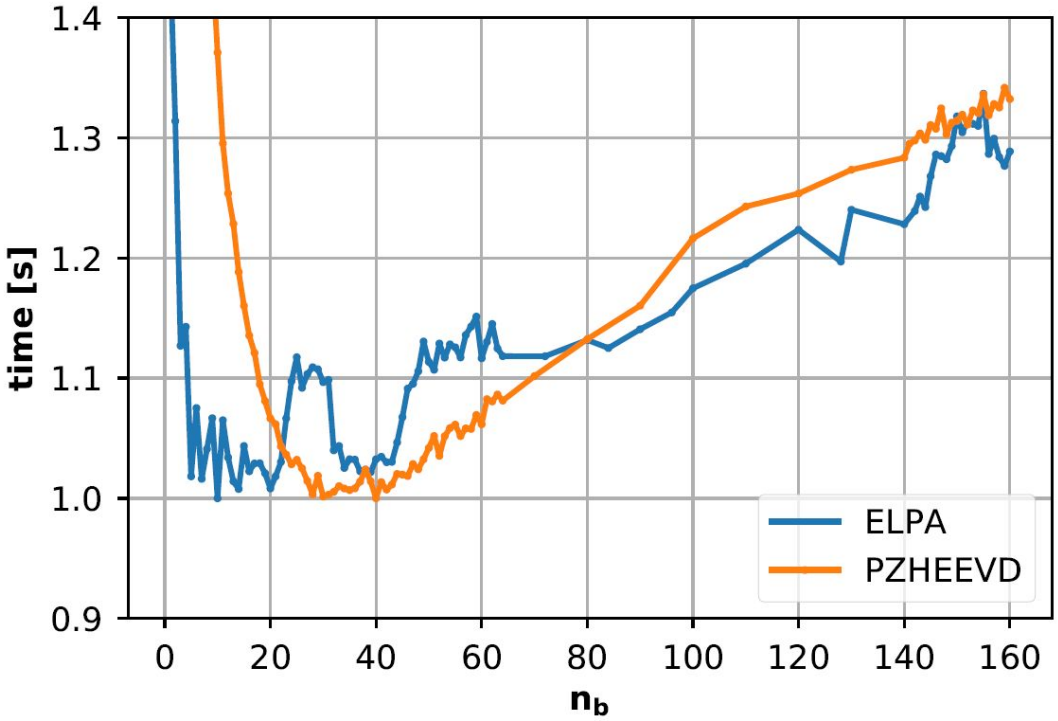
Distribution of performance



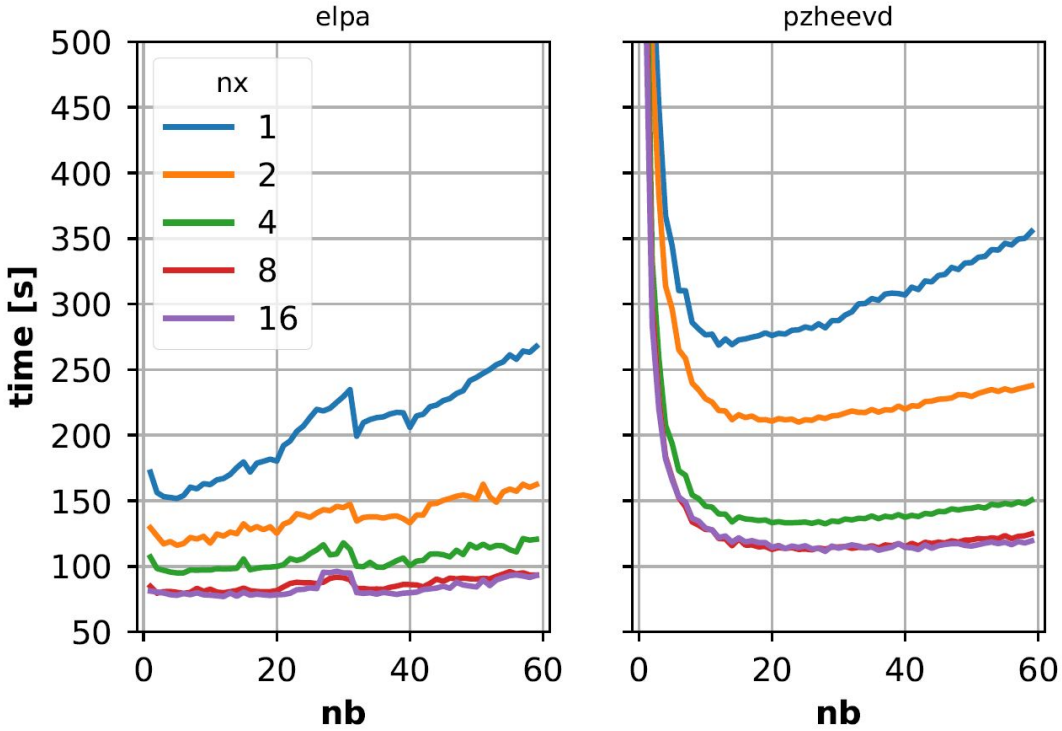
(nx,ny)	ALL
nb	1...100
N	20000
Architecture	KNL
Nodes	4

Parameter landscape - nb

(nx,ny)	(10, 32)
N	40000
Architecture	SKX
Nodes	8

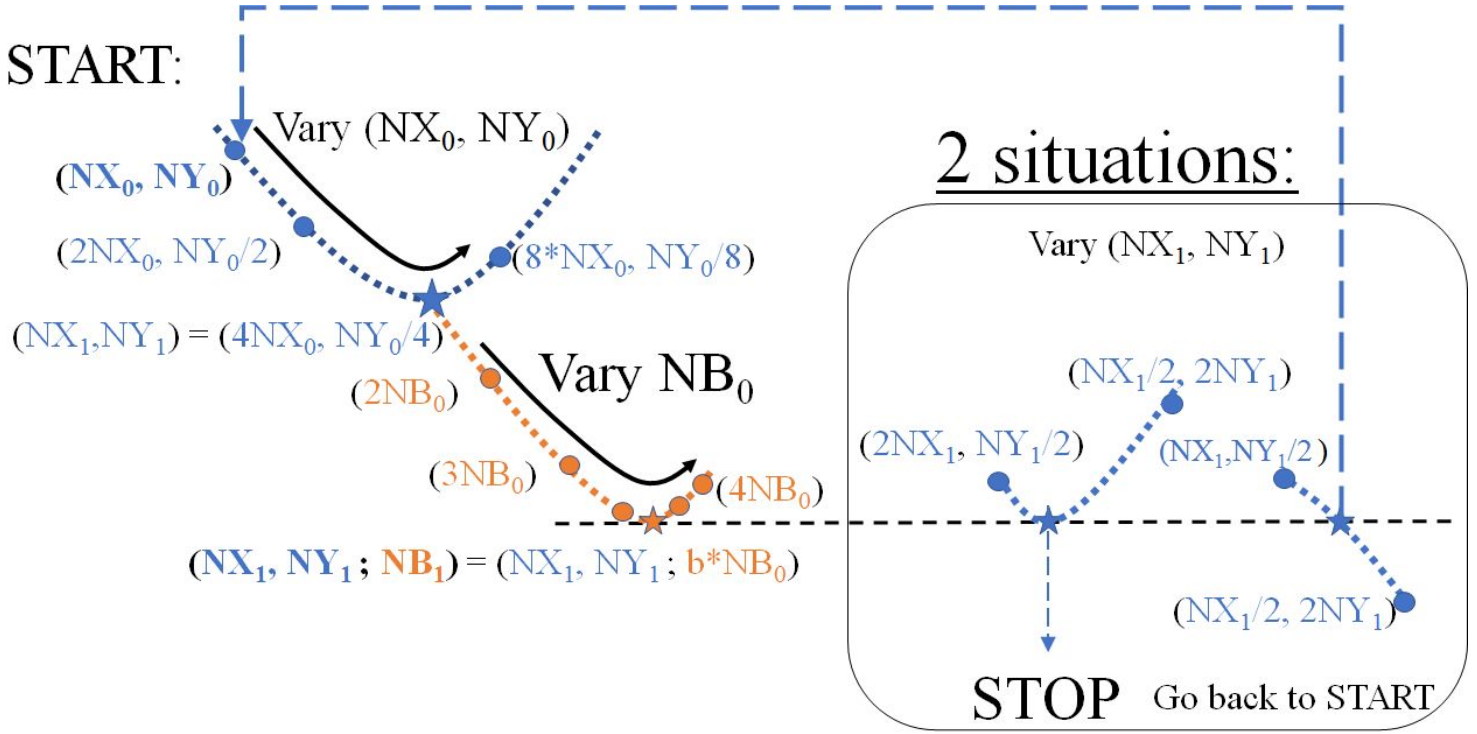


Parameter landscape

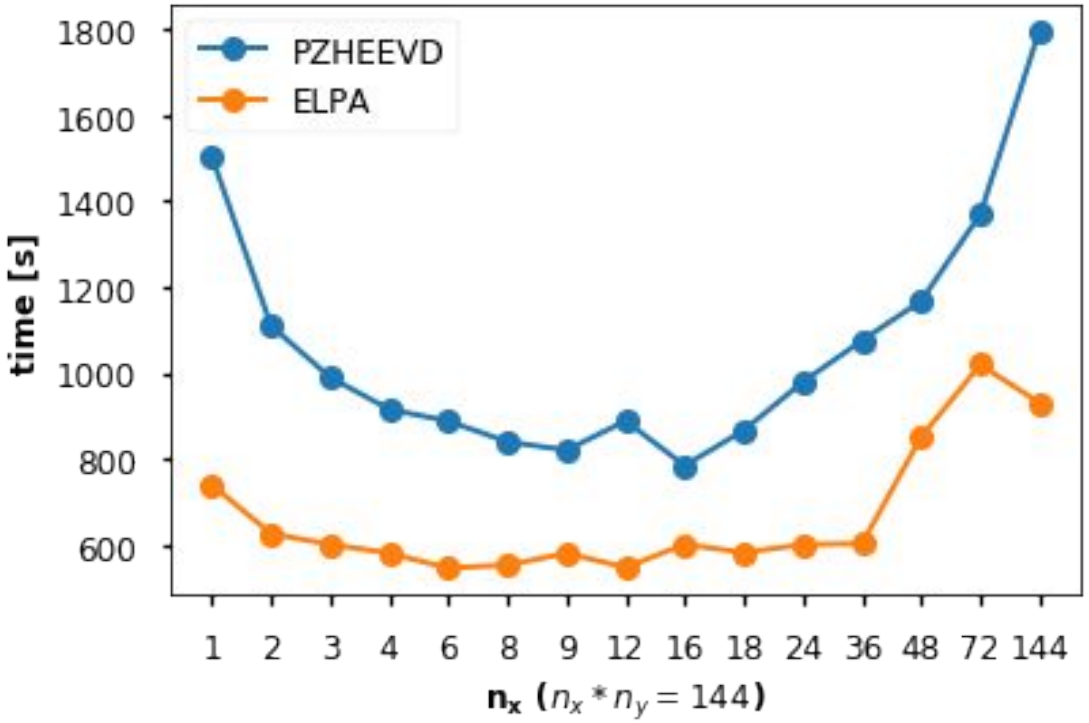


N	20000
Architecture	KNL
Nodes	4

Manual optimization



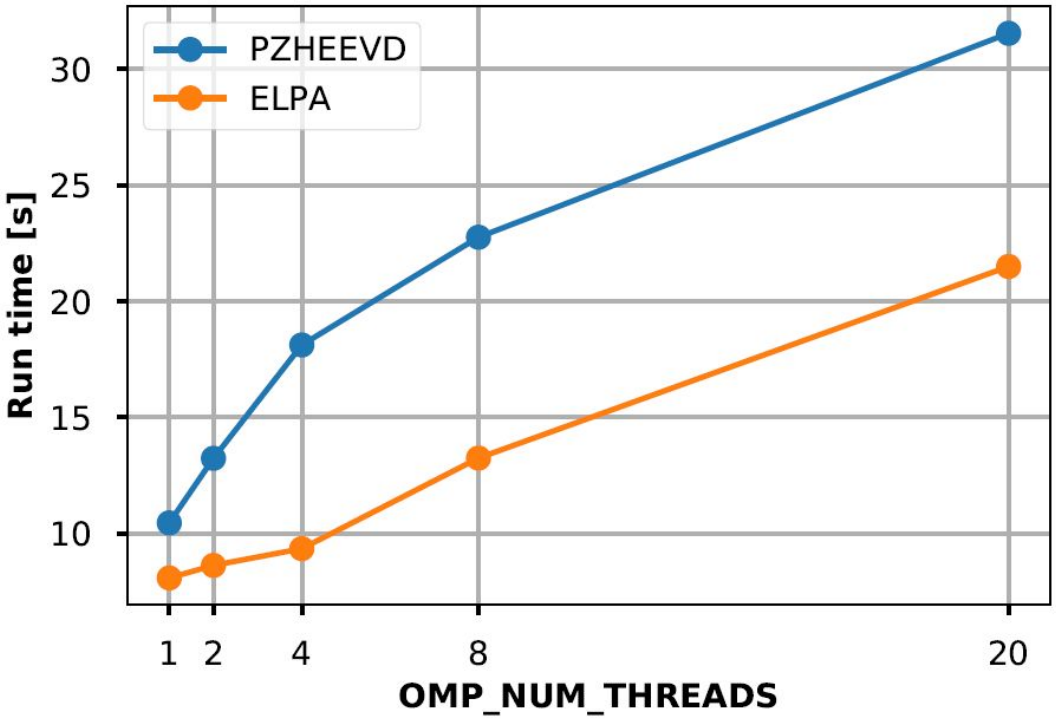
Optimal (nx,ny)



- nx=ny is good, but not the best
- (nx,ny) != (ny,nx)
- nx|ny = 1 is always bad

N	20000
Architecture	BDW
Nodes	4
MPI ranks	144

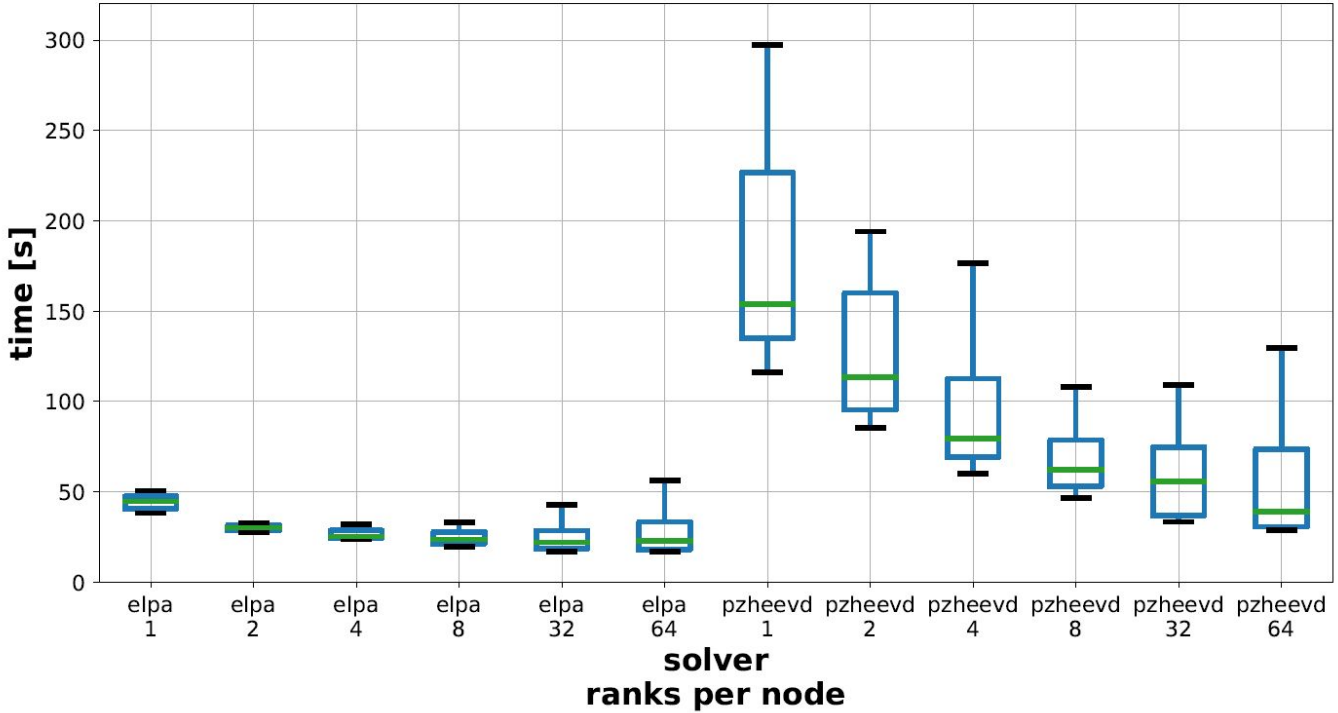
OpenMP scalability



- OMP_NUM_THREADS * MPI_Ranks kept constant
- No OpenMP penalty would be a horizontal line
- ELPA is OK up to 4 threads
- Useful for e.g. Quantum Espresso Exact Exchange simulations

(nx,ny)	best
nb	best
Architecture	SKX
nodes	4

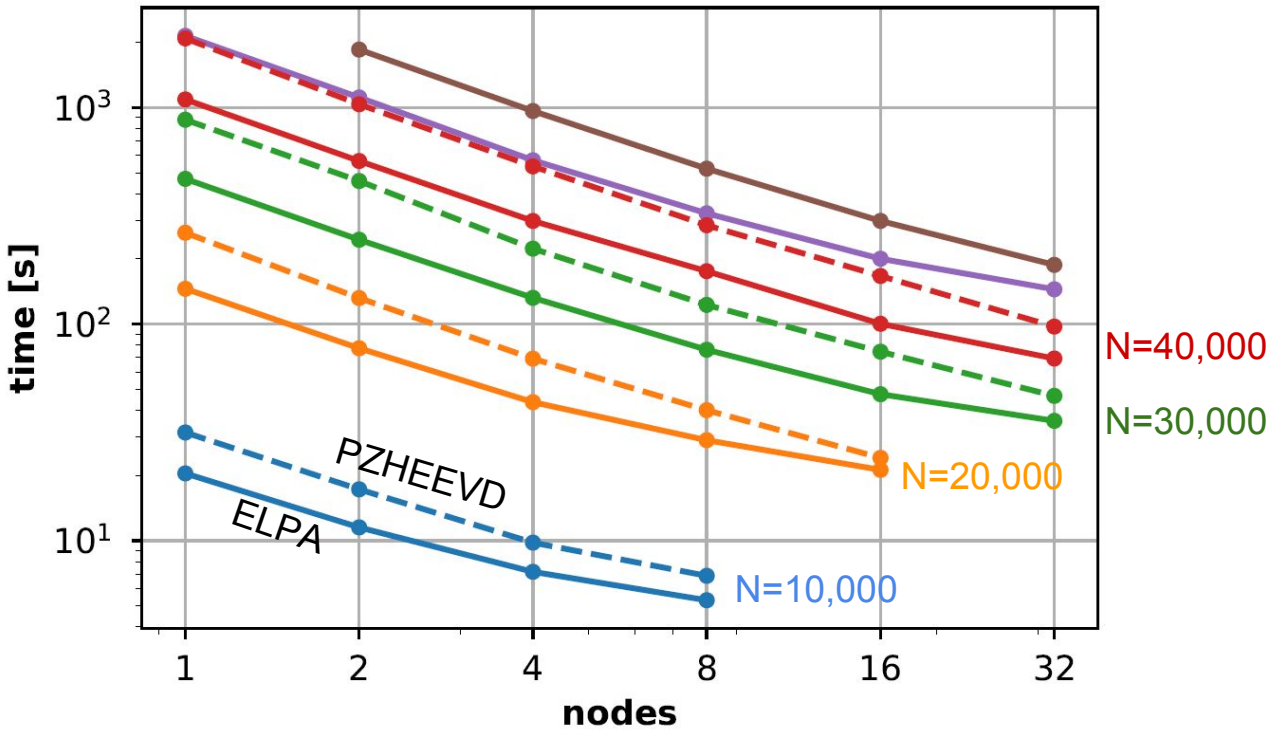
OpenMP performance



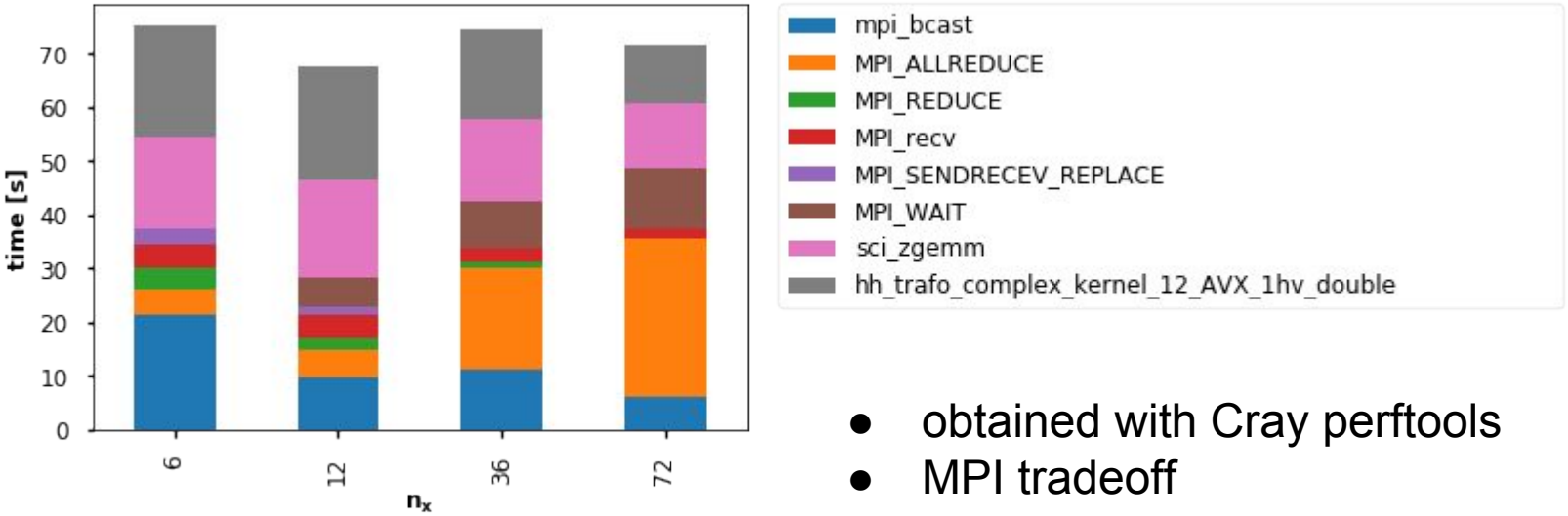
N	10000
Architecture	KNL
Nodes	4

Scaling of ELPA and PZHEEVD

(nx,ny)	best
nb	best
Architecture	SKX



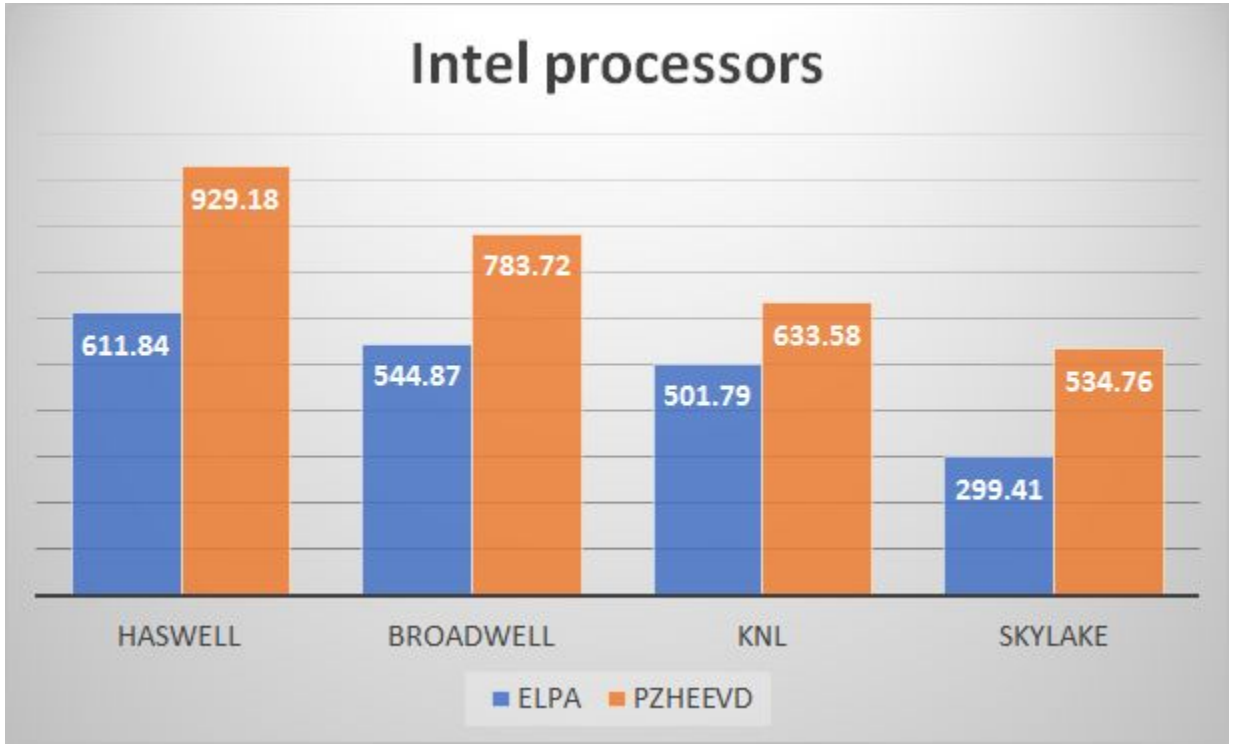
ELPA profile



- obtained with Cray perftools
- MPI tradeoff
- nx=12 is best overall time

Processor evolution

solution time on 4 nodes



Benchmarking programs

APPENDIX A.
ELPA BENCHMARK

```
program benchmark_elpa
use elpa
use elpa_driver
...
call blacs_get(...)
call blacs_gridinit(...)
call blacs_gridinfo(...)

...
e => elpa_allocate()
call e%set("na", ln, err)
...
call e%set("complex_kernel",
ELPA_2STAGE_COMPLEX_AVX512_BLOCK1, err)
call e%eigenvectors(a_sym, w, z, err)

end program benchmark_elpa
```

APPENDIX B.
PZHEEVD BENCHMARK

```
program benchmark_pzheevd
. . .


call blacs_get(...)
call blacs_gridinit(...)
call blacs_gridinfo(...)
. . .

call pzheevd(..., temp, ...)
. . .


call pzheevd(...,work, ...)

end program benchmark_elpa
```

Conclusions

- In general, ELPA is 20-30% faster than ScaLAPACK. When possible, choose ELPA (Quantum Espresso, VASP, etc).
- NX close to NY, is an optimum, but not always.
- NB: SCALAPACK choose a value in the 50s. ELPA choose a value in the 10s.
- The search space of ELPA contains multiple minima.
- The optimum (NX,NY) in ELPA is an interplay between MPI_Bcast and MPI_Allreduce.
- Threading is better in ELPA than in ScaLAPACK. More work needs to be done in that direction.
- Looking to expand this eigenvalue benchmarking suite:
  - FEAST
  - ELEMENTAL
  - ...

Extras

Spearmint optimization