IMAMURA Group (RIKEN AICS)

Development of a High Performance Dense Eigensolver: EigenExa

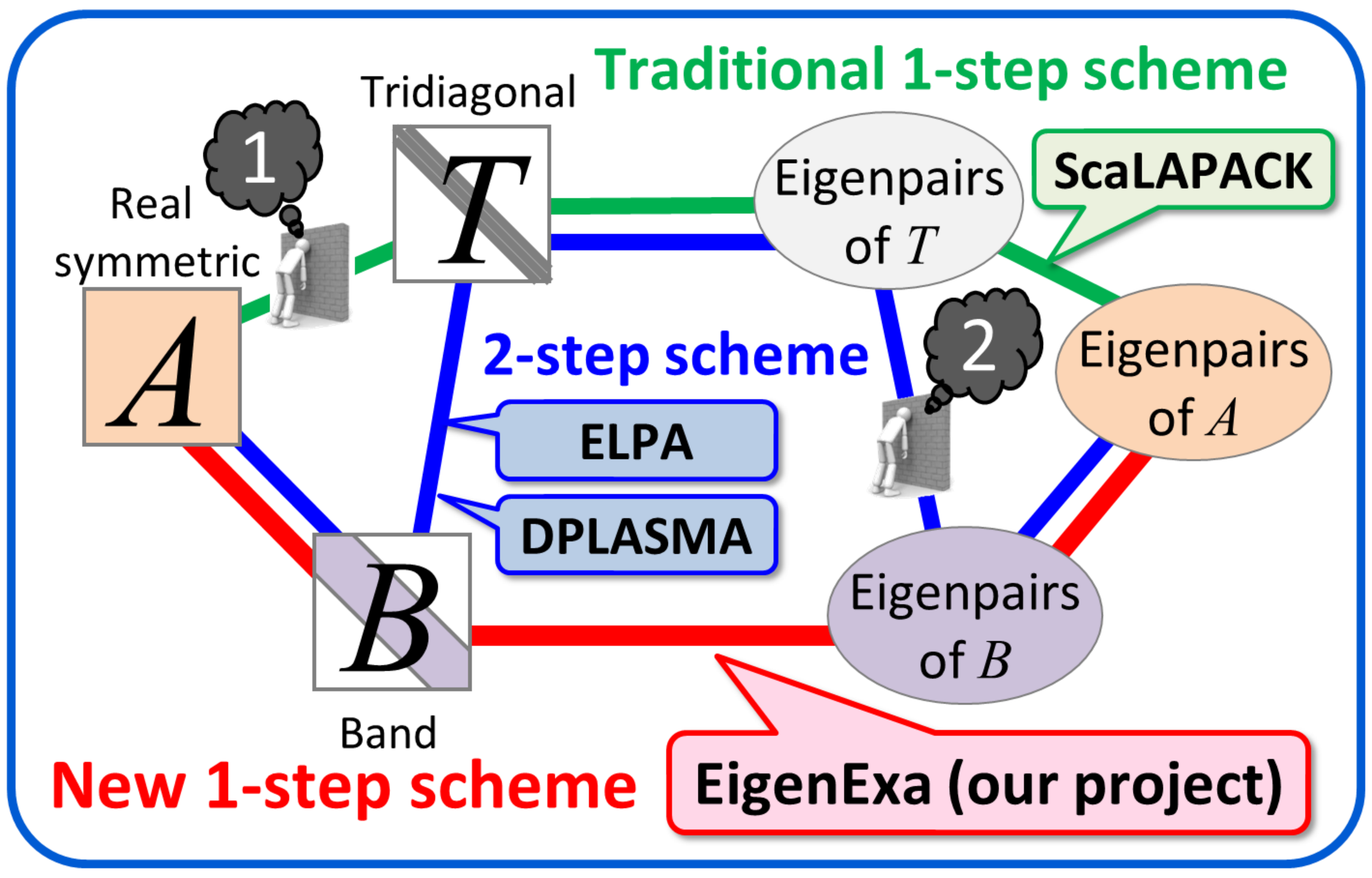
◆ **Features:** *overcoming performance bottlenecks by a new 1-step scheme*

(1) Bound by narrow memory band width

⇨ Reducing required B/F ratio by block algorithm

(2) Difficulty in high performance implementation of back transformation from tri. to band (critical when requiring all/many eigenpairs)

⇨ Avoiding by directly calculating the eigenpairs of the band matrix

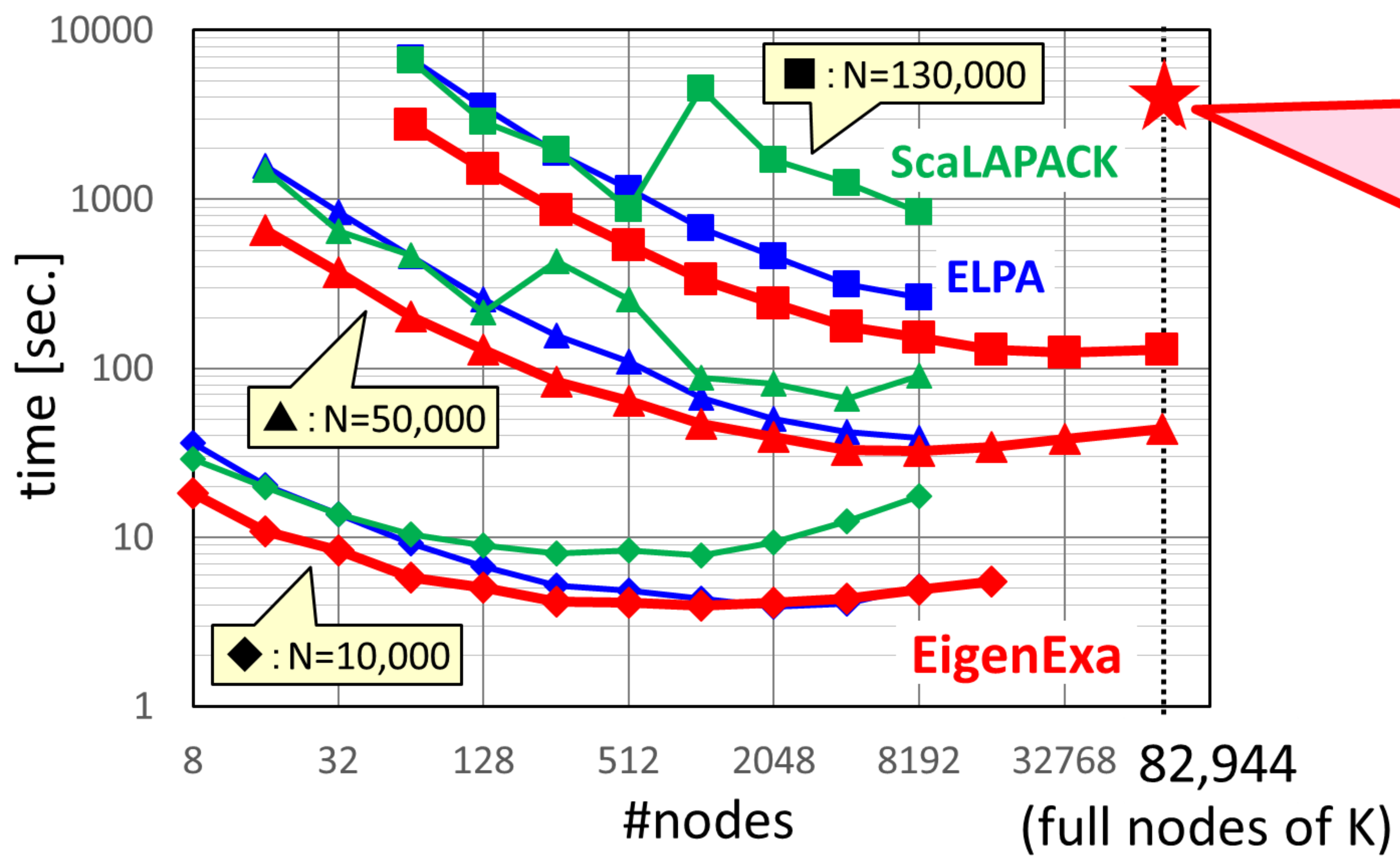


◆ Achievement

➢ Performance results on the K computer

Note : ELPA is less tuned for the K computer than other two programs.

total execution time



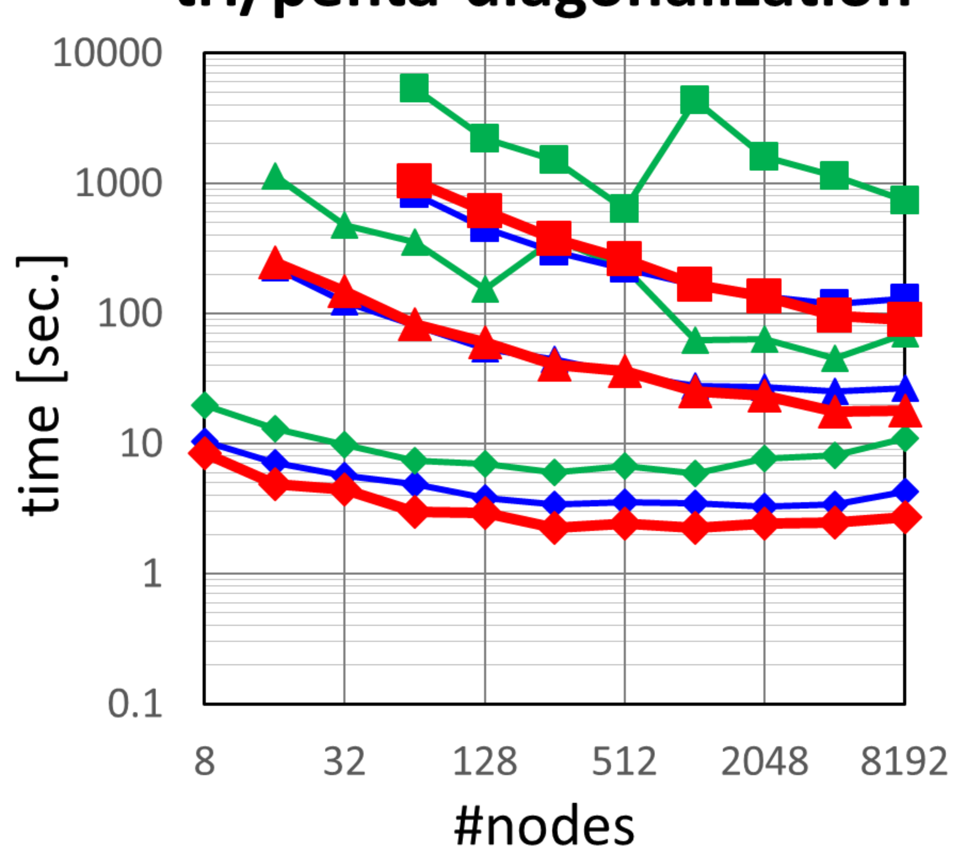
one million dimensional dense eigenvalue computing

- using full nodes of K (82,944 procs., 663,552 cores)
- 3,464 sec. (penta: 1,968s, D&C: 922s, back: 574s)
- 1.7 PFLOPS (16% of peak)
- Residual : $\max_i \|A\boldsymbol{v}_i - \lambda_i \boldsymbol{v}_i\|_2 / \|A\|_F = 3.1 \times 10^{-13}$
- Orthogonality : $\|V^\top V - I\|_F = 2.1 \times 10^{-10}$

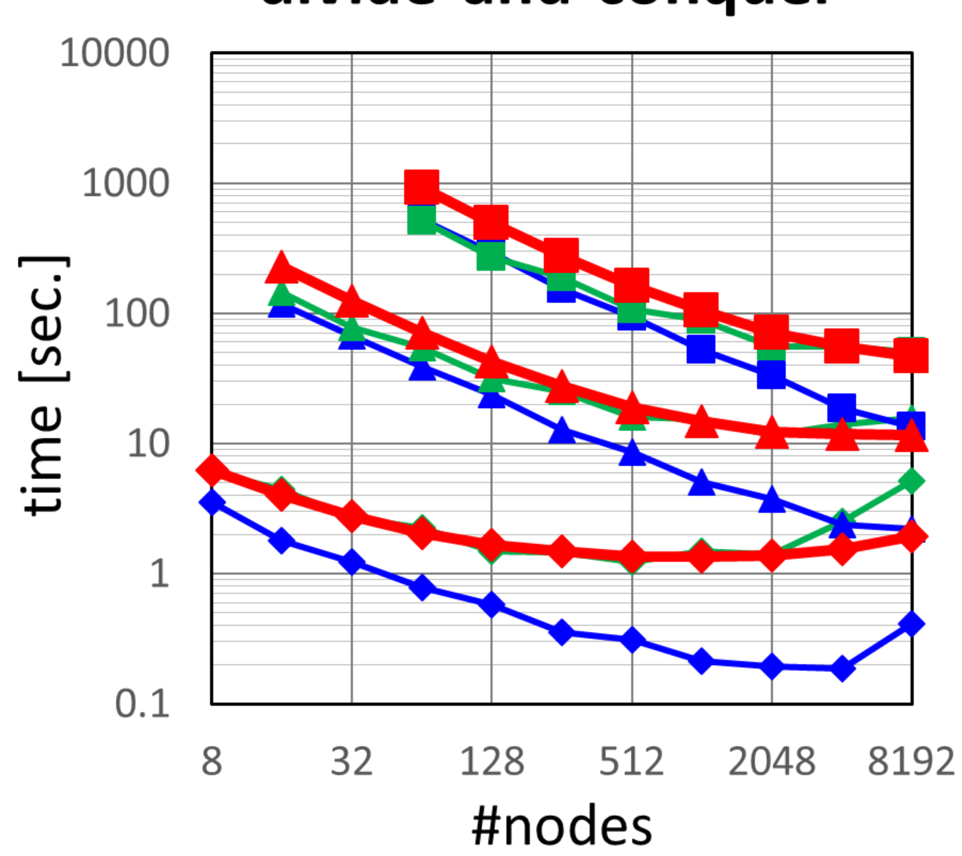
evaluation conditions

- Calculations : all eigenpairs of N x N random matrix
- Libraries : BLAS / MPI provided from Fujitsu on K
- Assignments : 1 MPI process with 8 threads / 1 node

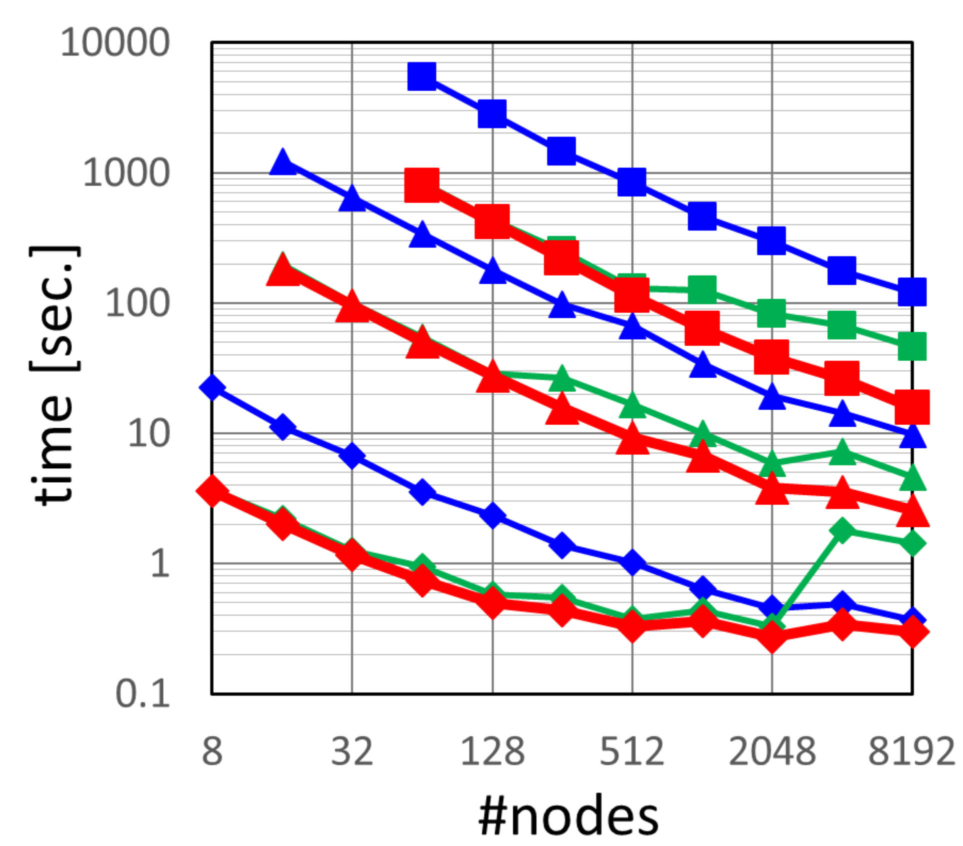
tri/penta-diagonalization



divide-and-conquer



back transformation



➢ Observation on EigenExa

- ✓ Penta-diagonalization : fast and scalable because of our original implementations
- ✓ Divide-and-conquer : slow and not scalable since just ported from ScaLAPACK
- ✓ Back transformation : much faster than ELPA due to employing the 1-step scheme

◆ Future work

- ✓ Improvement by communication-avoiding/hiding implementations
- ✓ Development for GPU-accelerated/many-core environments
- ✓ Investigation into and redevelopment of the divide-and-conquer routines

EigenExa latest ver. is available from ***http://www.aics.riken.jp/labs/lpnctrt/EigenExa_e.html***