**OVERVIEW**
Considered one of the top 10 algorithms of the 20th century, the Fast Fourier Transform (FFT) is widely used by applications in science and engineering. Large scale parallel applications targeting exascale, such as those part of the Exascale Computing Project (ECP), are designed for heterogeneous architectures and, currently, some of them rely on efficient state-of-the-art FFT libraries built as CPU kernels. We have developed and released heFFTe \[1\], a hybrid highly-scalable and robust library for multidimensional FFT computations targeting exascale.

**PERFORMANCE AND RESULTS**

**A. Speedup for Local Kernels Computation**
Typical local kernels for packing, unpacking and computation are available on CPU based state-of-the-art libraries, heFFTe provides new GPU kernels for these tasks achieving over 40x speedup.

![Fig 3. FFT of size 1024x1024x1024 on 4 nodes, 32 MPIs per nodes (CPU kernels, left) vs. 24 MPI processes, 6 MPIs (6 Volta100 GPUs) per node (GPU kernels, right).](image)

**B. Scalable Performance**
The GPU version of heFFTe has very good weak and strong scalability, and achieves close to 90% of the roof-line theoretical peak performance.

![Fig 4. Strong (left) and weak (right) scalability. We use 24 MPIs/node on each case, 1MPI/core for heFFTe CPU and 4MPI/GPU-Volta100 for heFFTe GPU.](image)

**C. Using heFFTe in ECP Applications**
Applications such as LAMMPS (EXAALEP-ECP) rely on their own FFT library (FFTMPI for this case). We provide wrappers to directly call heFFTe and observe the performance gain, while maintaining good scalability \[2\].

![Fig 5. Rhodopsin protein benchmark for LAMMPS, on 2 nodes and 4 MPI ranks per node, on a 128x128x128 FFT grid.](image)

**REFERENCES**

- [1] https://bitbucket.org/icl/heffte/