Accelerating the Fast Fourier Transform on Large Scale Heterogeneous Systems

"The heFFTe Library"

Ahmad Abdelfattah
Innovative Computing Laboratory
University of Tennessee, Knoxville

MUG'24, Columbus, OH
August 19-21, 2024

Credit to the heFFTe team

- Miroslav Stoyanov (ORNL)
- Alan Ayala (UTK → AMD) – yesterday's talk
- Stan Tomov (UTK → NVIDIA)
- Azzam Haidar (UTK → NVIDIA)
- Jack Dongarra (UTK)
- Sebastien Cayrols (UTK → NVIDIA)
- Jiali Li (UTK)
- George Bosilca (UTK → NVIDIA)
- Veronica Montanaro (ETH)
- Sonali Mayani (ETH)
- Andreas Adelmann (ETH)
- students and outside collaborators

➢ I am the newest member to this team ☺

The purpose of this talk

1) High-level introduction to heFFTe as an efficient distributed implementation of multi-dimensional FFT
2) The status of heFFTe post ECP
3) heFFTe as a benchmark for MPI implementations
4) Impact on some applications
5) Future Directions

Fast Fourier Transform (FFT)

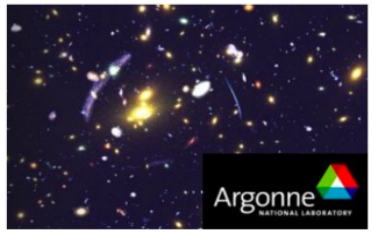
- FFT computes the Discrete Fourier Transform (DFT) of a series:
  Let $x = x_0, \ldots, x_{N-1}$ are complex numbers. The DFT of x is the sequence $X = X_0, X_1, \ldots, X_{n-1}$,such that:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i 2\pi k n / N} \qquad k = 0, \ldots, N-1.$$

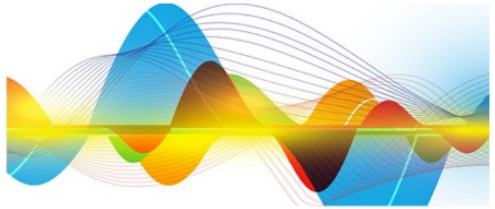
- DFT can be computed as a matrix-vector multiplication (GEMV) in $\mathcal{O}(N^2)$ FLOPs (memory-bound)
- FFT reduces the complexity to $\mathcal{O}(N \log_2 N)$ (even more memory-bound)
- The Inverse Discrete Fourier Transform (IDFT) is similarly defined except that the *'e'* exponents are taken as $(i\, 2\pi\, k\, n\, /\, N)$, and elements divided by N

FFT in the DOE's Exascale Computing Project (ECP)

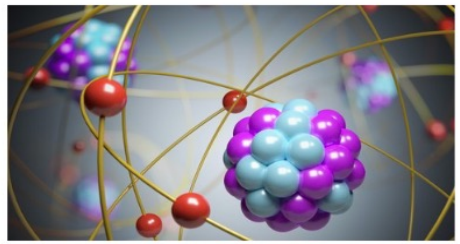
Cosmology
***ECP* ExaSky - HACC**



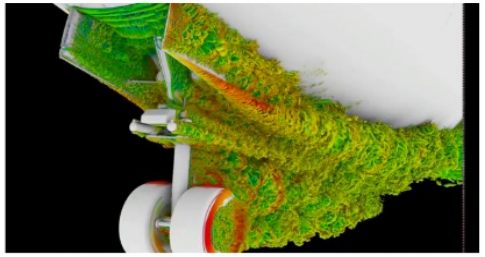
Signal processing,
***ECP* WARPX**



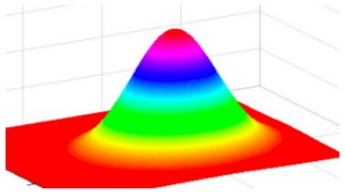
Deep Learning



Molecular Dynamics
***ECP* EXAALT**



Particle Simulations
***ECP* CoPa / Cabana**



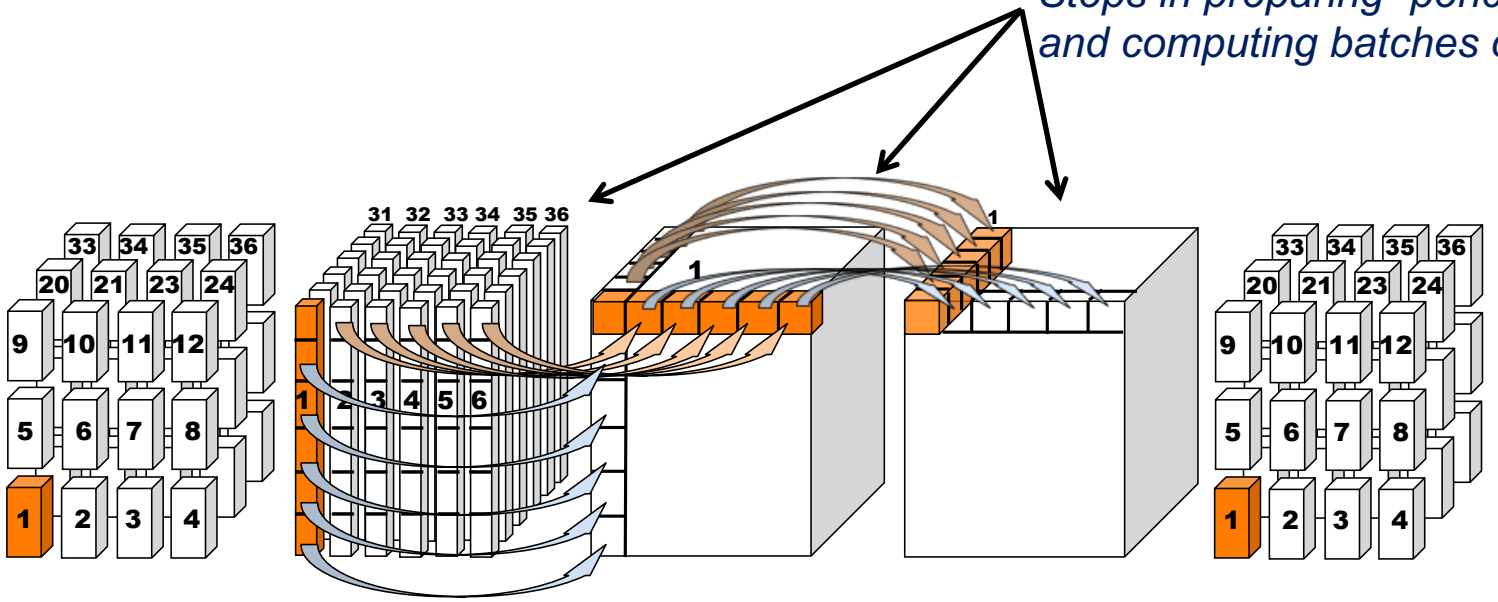
PDE solutions, **MASSIF**

- Some of these applications require multi-dimensional FFT at large scale
- Must run on DOE's ECP systems (i.e. use GPUs)

heFFTe: Highly Efficient Exascale FFTs Library for Heterogeneous Architectures

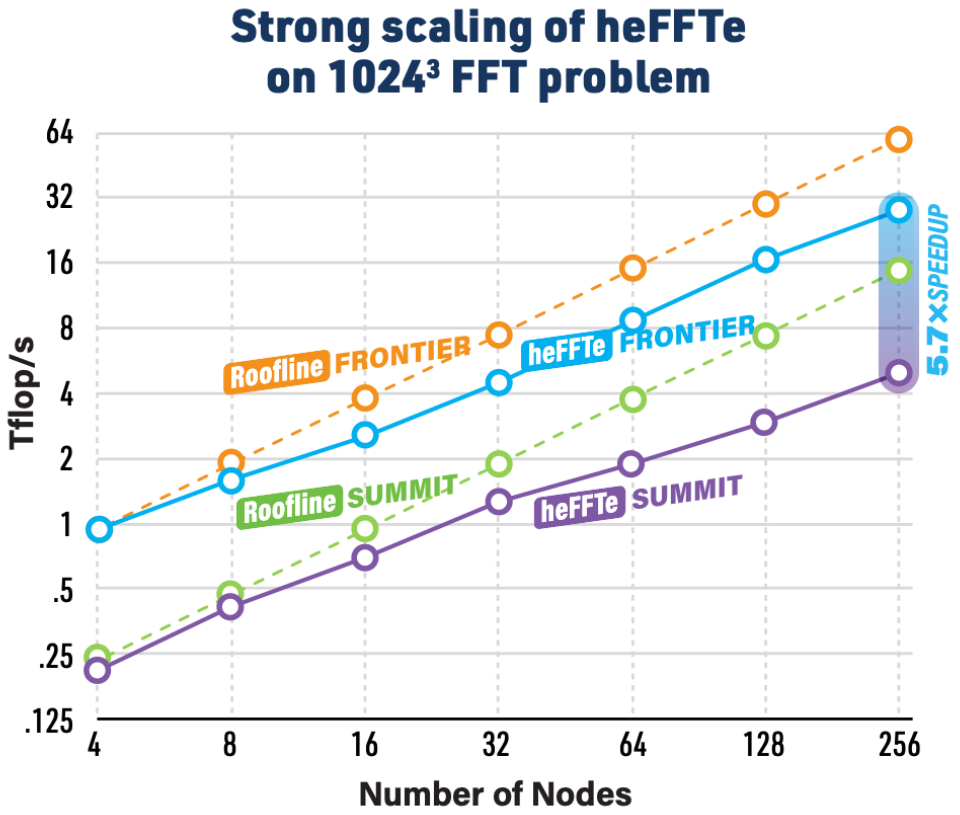
- 1D, 2D, and 3D distributed FFT library
  - GPU-enabled
  - Relies on single-node FFT libraries (FFTW, cuFFT, rocFFT, & oneMKL)
  - Simple data transposition/reshape kernels
  - MPI Communication

Steps in preparing "pencils" in the different dimensions and computing batches of 1D FFTs locally on each node/GPU



Status of heFFTe

- heFFTe-2.4.0 with support for CPUs, Nvidia, AMD, and Intel GPUs
- Capabilities:
  - Multidimensional FFTs
  - C2C, R2C, C2R
  - DCS, DST, and convolution
  - Batched FFTs
  - Multiple data layouts & communications patterns
- Open-source Software
  - Spack installation and integration in xSDK
  - Homepage: http://icl.utk.edu/fft/
    Repository: https://github.com/icl-utk-edu/heffte
- So far, no major developments after ECP closeout



heFFTe among Other Developments

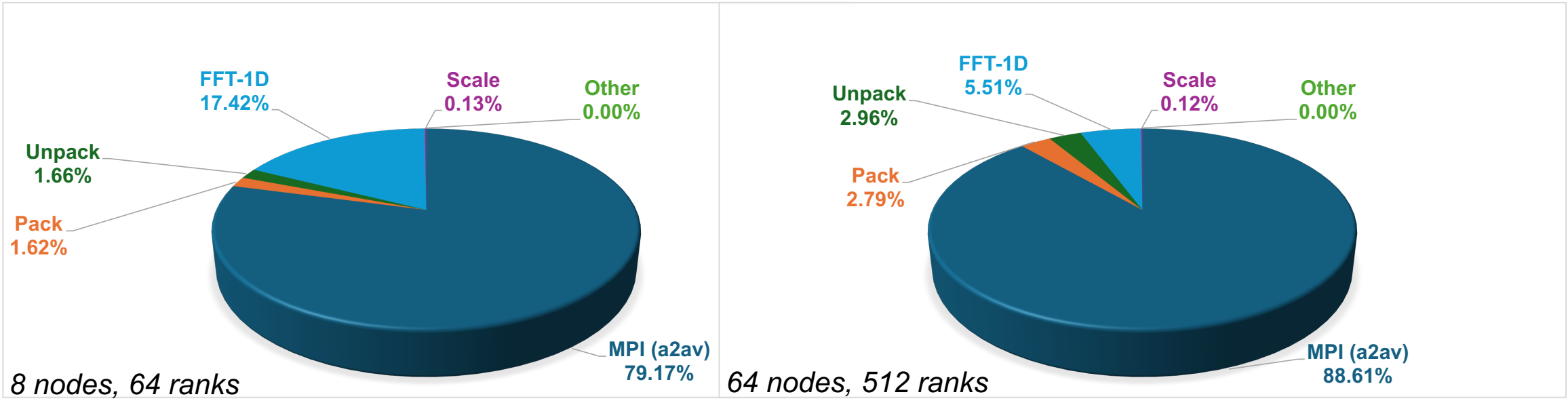
- Amongst the very few parallel FFT libraries that support GPUs, heFFTe provides unique functionalities that cover a large number of features from the state-of-the-art, making it ubiquitous for a wide range of applications

	Library	Pencil Decomp	Brick Decomp	Slab Decomp	Transpose Reshape	Stride Reshape	R2C Transform	Single precision	Mixed precision	Multiple backends	Nonblocking All-to-All
CPU	FFTW3	✓				✓	✓	✓			
CPU	FFTMPI	✓	✓		✓			✓		✓	
CPU	2DECOMP	✓				✓	✓				
CPU	SWFFT		✓		✓						
CPU	PFFT	✓			✓		✓				
CPU	P3DFFT	✓		✓	✓		✓	✓			✓
GPU	AccFFT	✓			✓	✓	✓	✓		✓	
GPU	FFTE	✓		✓	✓		✓	✓			
GPU	heFFTe	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

heFFTe is heavily communication-bound

- Using heFFTe's own minimal tracing tool
- MPI calls dominate the execution time, especially on the GPU
- Any improvement in communication leads to huge performance gains

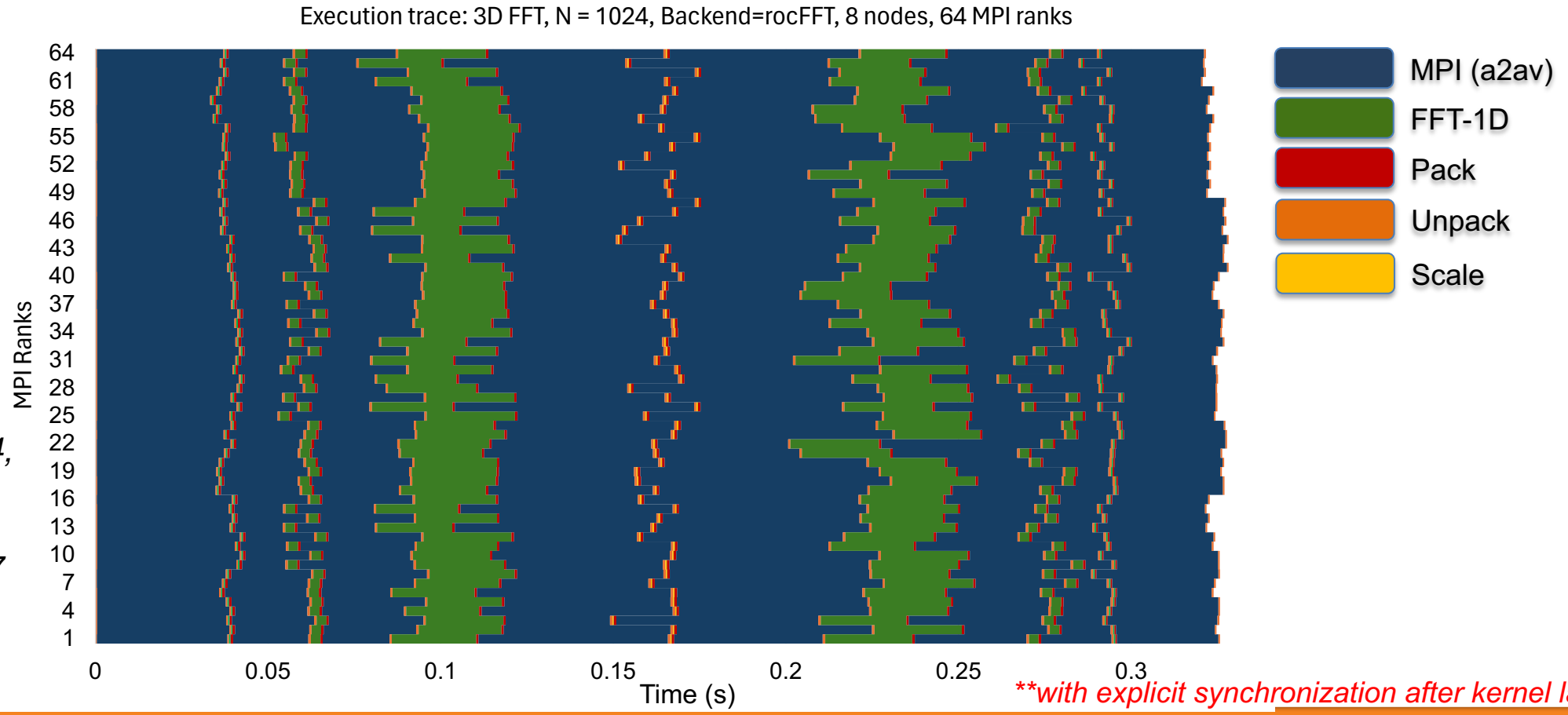
Time Breakdown of heFFTe on Frontier using rocFFT
3D FFT, N = 1024, FP64 (C2C), ROCm-5.6.0, cray-mpich/8.1.27



***with explicit synchronization after kernel launches*

heFFTe is heavily communication-bound

- Using heFFTe's own minimal tracing tool
- MPI calls dominate the execution time, especially on the GPU
- Any improvement in communication leads to huge performance gains

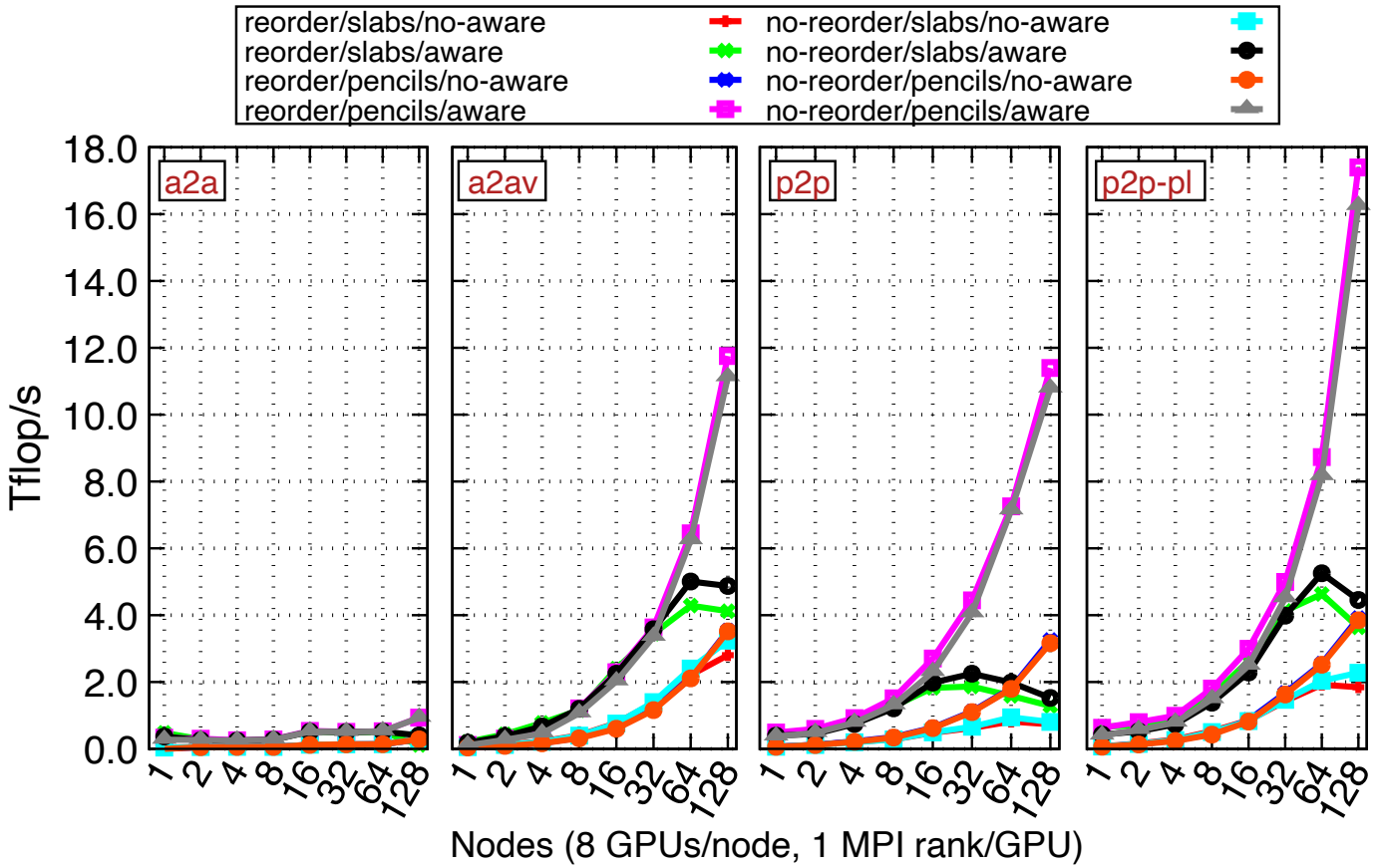


3D FFT, N = 1024,
FP64,
ROCm-5.6.0,
cray-mpich/8.1.27
8 nodes
64 ranks

***with explicit synchronization after kernel launches*

heFFTe benchmark sweep on Frontier

- 3D FFT, N = 1024
- rocFFT backend (ROCm-5.3.0)
- Cray-mpich-8.1.23
- 32 different runs
  - 4 communication patterns (a2a, a2av, p2p, p2p_pl)
  - 2 decompositions (pencils vs. slabs)
  - 2 FFT-1D modes (contiguous vs. strided)
  - 2 modes for MPI (std vs. GPU-aware)
- Could test/expose several aspects of an MPI implementation



- *p2p: uses MPI_Send and MPI_Irecv, receive is pipelined with packing and sending*
- *p2p_pl: uses MPI_Isend and MPI_Irecv, all sending receiving packing and unpacking are pipelined*

Integration to ECP EXAALT

LAMMPS Rhodopsin Benchmark using heFFTe

- Molecular dynamics apps heavily rely on FFTs, and often have their own parallel FFT implementation (e.g., fftMPI, SWFFT).
- Using heFFTe real-to-complex accelerates LAMMPS Kspace kernel around 1.76×.

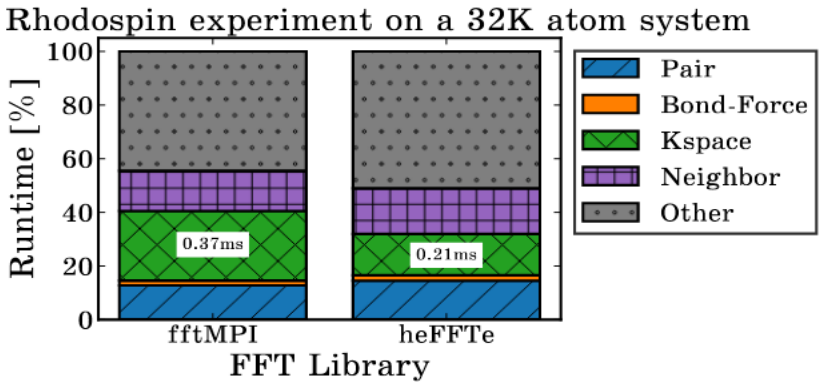


Figure: Breakdown for the LAMMPS Rhodopsin experiment. Using 32 Summit nodes, 6 V-100 GPUs per node, and 1 MPI per GPU.

Ref.: Performance Analysis of Parallel FFT on Large Multi-GPU Systems.

Ayala et al., IEEE IPDPS 2022.

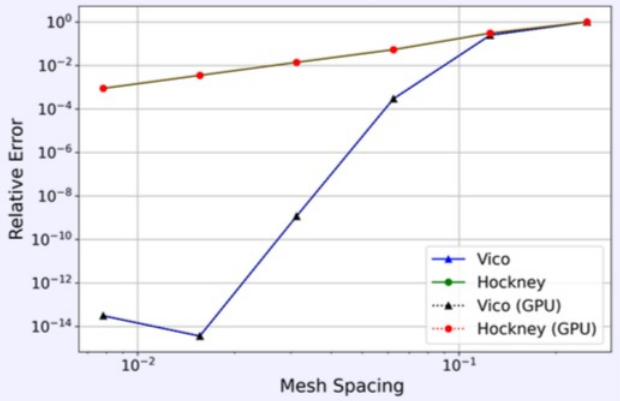
Opportunity: Approximate FFT

- Some applications tolerate accuracy in FFT
- approximate FFT computations (with casting to FP32 / FP16)

Novel Poisson solver: Problem

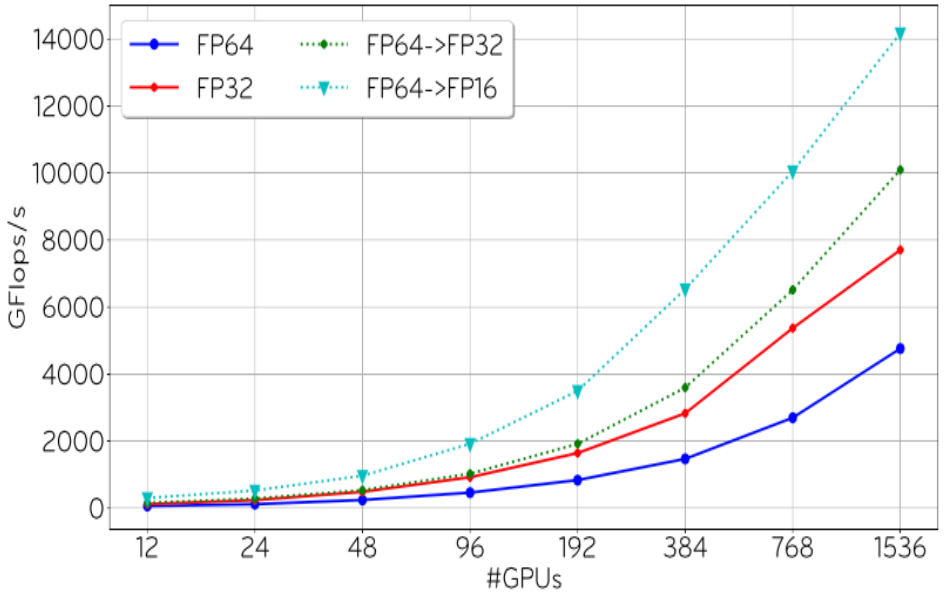
What do we gain?

Convergence for smooth Gaussian source:



- These solvers are in Independent Parallel Particle Layer (IPPL) and require Discrete Cosine Transform of type 1
  Montanaro et al. (ETH)

Impact on Performance



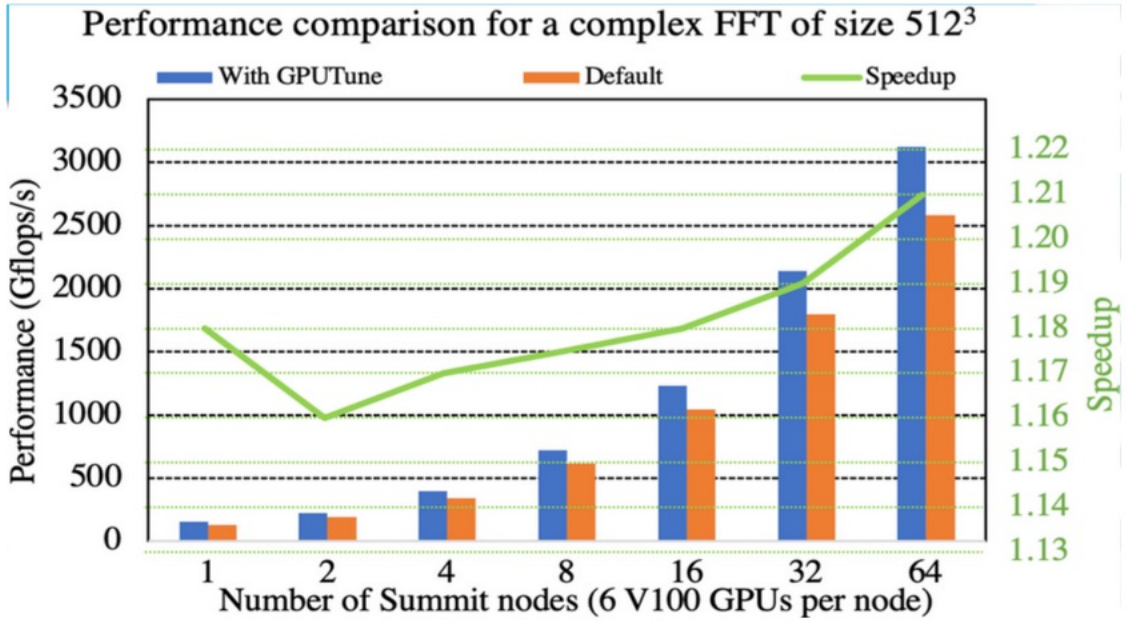
Impact on Accuracy

#GPU	FP64	FP32	FP64 → FP32
12	6.00e-15	4.96e-06	1.94e-07
24	6.17e-15	4.91e-06	2.20e-07
48	5.92e-15	4.49e-06	3.01e-07
96	6.00e-15	3.47e-06	3.90e-07
192	5.11e-15	3.54e-06	3.99e-07
384	5.25e-15	4.44e-06	5.09e-07
768	5.29e-15	3.13e-06	5.44e-07
1536	5.38e-15	3.06e-06	5.57e-07

heFFTe Tunability and Configurability

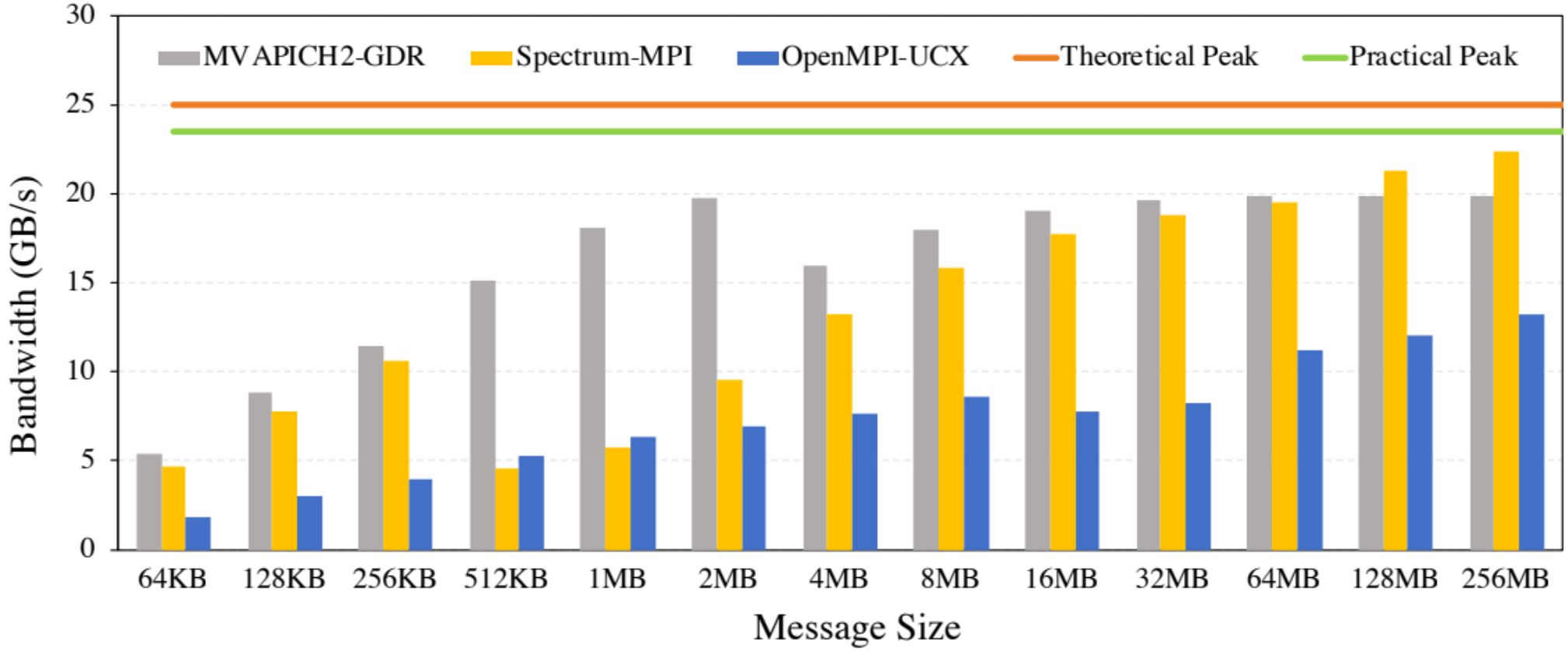
- Which set of options is best for a given workload?
  - Multiple MPI implementations (a2a, a2av, p2p, pipelined p2p)
  - Multiple decompositions (pencils, slabs)
  - Multiple layouts (contiguous, strided)
  - Run-time parameters ☺

- Auto-tuning heFFTe using GPTune (https://gptune.lbl.gov/), we were able to increase performance by tuning FFT input parameters and communication settings
- Shown is performance improvements and speedup on Summit (~15 - 20%)



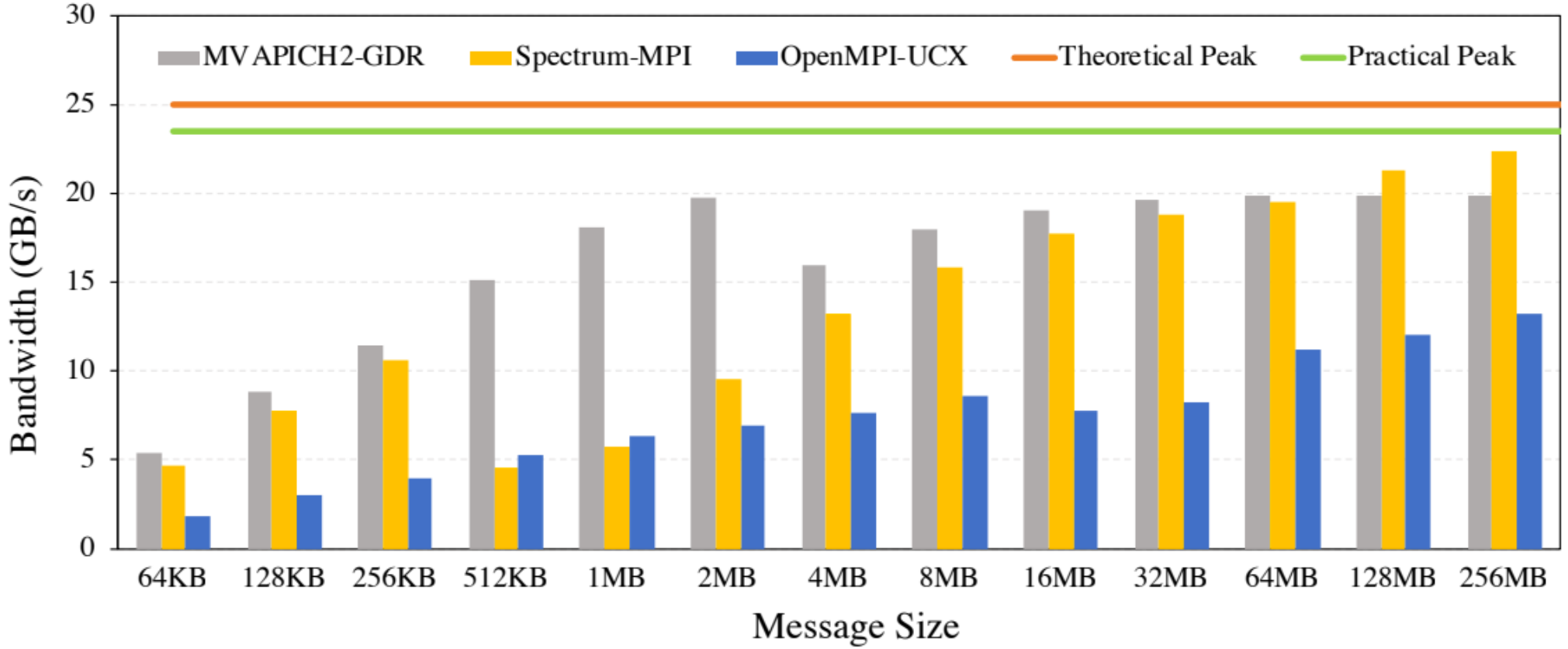
heFFTe using MVAPICH

Bandwidth benchmark for several MPI implementations on Summit (16 nodes)



heFFTe using MVAPICH

Bandwidth benchmark for several MPI implementations on Summit (16 nodes)



- > 50% improvement in latency and bandwidth against Cray-mpich on Frontier (AlltoAll) -- (Dr. Panda's keynote yesterday)
- More expected with on-the-fly data compression

Conclusion

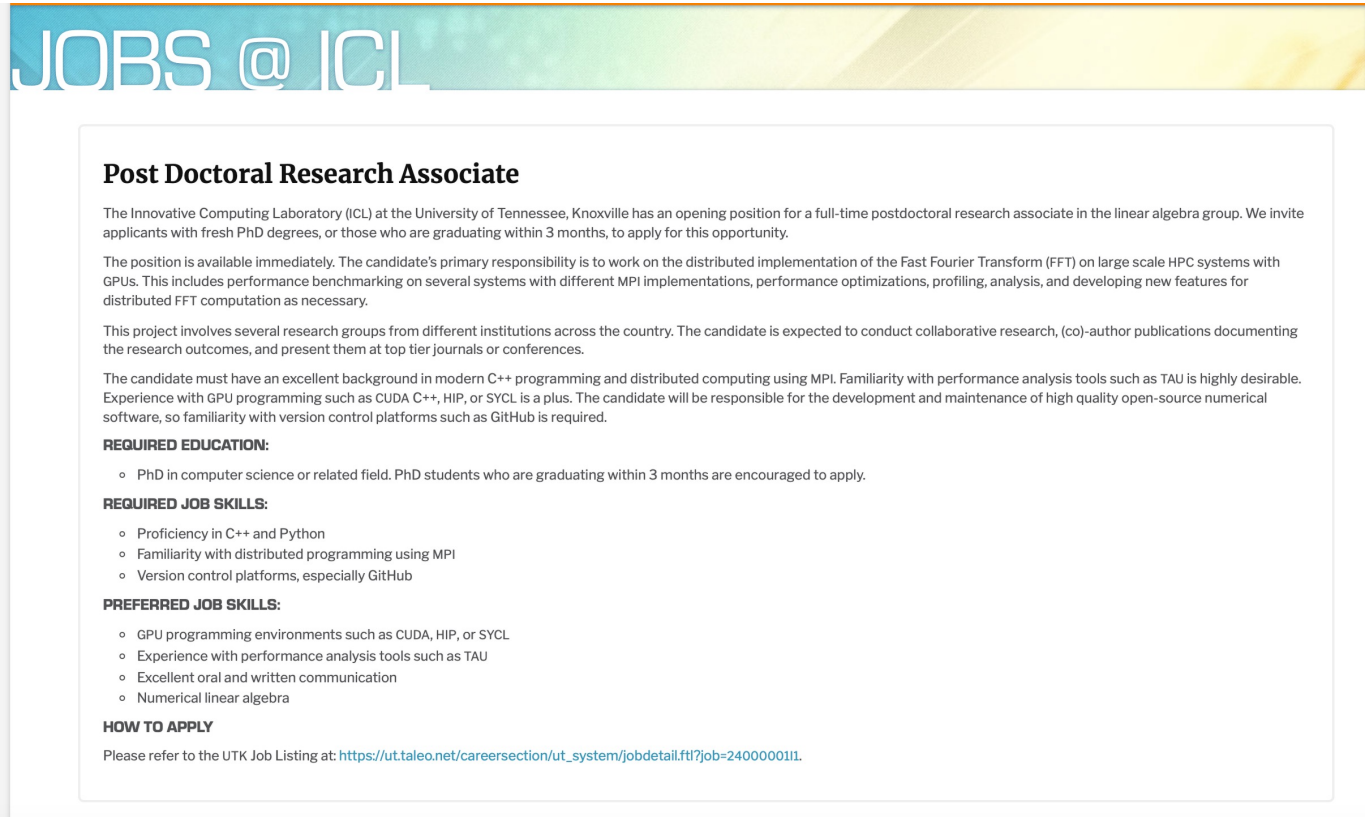
- heFFTe is an ECP-funded library for multi-dimensional FFT computations
  - Mainly targeting DOE's Exascale system
  - GPU-enabled for NVIDIA, AMD, and Intel GPUs
  - Highly configurable
- heFFTe can serve as a good benchmark for MPI implementations
  - ~90% of execution time is spent in MPI calls
  - Uses different communication patterns (a2a, a2av, p2p, and pipelined p2p)

Future Directions

- heFFTe using MVAPICH-Plus 4.0b
  - Joint project with Dr. Panda's group to use MVAPICH as UMS on Frontier
    - MVP-4.0b brings promising improvement over Cray-mpich (> 50% for latency & BW in AlltoAll)
    - On-the-fly compression could yield even more significant performance gains
  - Plan to target a large NVIDIA system based on H100 (MN5 @ BSC)
- Robust profiling and tracing with TAU
  - Expose potential bottlenecks in heFFTe or the underlying MPI implementation
- Auto-tuning framework for heFFTe?

We are hiring!

- Position for a postdoctoral research associate
- https://icl.utk.edu/jobs/



JOBS @ ICL

Post Doctoral Research Associate

The Innovative Computing Laboratory (ICL) at the University of Tennessee, Knoxville has an opening position for a full-time postdoctoral research associate in the linear algebra group. We invite applicants with fresh PhD degrees, or those who are graduating within 3 months, to apply for this opportunity.

The position is available immediately. The candidate's primary responsibility is to work on the distributed implementation of the Fast Fourier Transform (FFT) on large scale HPC systems with GPUs. This includes performance benchmarking on several systems with different MPI implementations, performance optimizations, profiling, analysis, and developing new features for distributed FFT computation as necessary.

This project involves several research groups from different institutions across the country. The candidate is expected to conduct collaborative research, (co)-author publications documenting the research outcomes, and present them at top tier journals or conferences.

The candidate must have an excellent background in modern C++ programming and distributed computing using MPI. Familiarity with performance analysis tools such as TAU is highly desirable. Experience with GPU programming such as CUDA C++, HIP, or SYCL is a plus. The candidate will be responsible for the development and maintenance of high quality open-source numerical software, so familiarity with version control platforms such as GitHub is required.

REQUIRED EDUCATION:

- PhD in computer science or related field. PhD students who are graduating within 3 months are encouraged to apply.

REQUIRED JOB SKILLS:

- Proficiency in C++ and Python
- Familiarity with distributed programming using MPI
- Version control platforms, especially GitHub

PREFERRED JOB SKILLS:

- GPU programming environments such as CUDA, HIP, or SYCL
- Experience with performance analysis tools such as TAU
- Excellent oral and written communication
- Numerical linear algebra

HOW TO APPLY

Please refer to the UTK Job Listing at: https://ut.taleo.net/careersection/ut_system/jobdetail.ftl?job=24000001l1.

Thank You


ICL
INNOVATIVE
COMPUTING LABORATORY


THE UNIVERSITY OF
TENNESSEE
KNOXVILLE