Poisson's equation in free space

1,000s of lines of code, cross-domain, cross-technology, cross-standenware, cross-platform, cross-algorithm.

Our approach

Have you ever wondered about this?

No analogue to LAPACK for spectral methods

- Medium-scale FFT (~100 data points) in most common library application but just 10x performance with open-source FFT tools, not full
- Higher-level FFTs rarely used

- Poisson solver in a plasma that uses FFT for the electric potential
- Some arithmetic intensity and scalability of FFTs can make this approach feasible

Issue 1: 2D FFT cell is standard kernel for many applications

- Problem sizes and applications reduce to 2D FFT cell
- Parallelism across dimensions and these are the challenges
- Supported by modern languages and environments: Python, Matlab...

Issue 2: FFTX is slowly becoming obsolete

- 2D 1.5-CPU in use, 1997
- FFTW 3.3 (2004) minor updates since then
- Need of high-performance FFT standard library

- Development currently dormant, except for small bug fixes
- No active support for accelerators (GPUs, x86 PMTs, GPGPUs)

- Parallel/MP assembly does not scale beyond 32 nodes

FFT is de-facto standard interface for FFT

- FFTX 2.6 is the high-performance reference implementation

- Vendor libraries support the FFTX 2.6 interface

- FFTW, OpenBLAS, ANL, AMD, and ACCA JPL of JPL, Intel, Sun, CRAY Library/CESM/FFTW

FFT: FFTX revamped for Exascale

Modernized FFTX-style interface

- Backends compatible to FFTX 2.6 and 3.x

- New FFTX kernels and parallel and vectorized implementations

- Small number of new features

- Faster computation efficiency, of blending, data placement, callback kernels

Code generation backend using SPIRAL

- Library/application kernels are interpreted as specifications in DSL

- Extract semantics from source code and known library semantics

- Compilation and advanced performance optimization across new library specification, accelerator development...

- Fine control over resource expenditure of optimization compute time, initialization time, execution time, optimization resources

- Reference library implementation and bindings to vendor libraries

- Library-user reference implementation for ease of development

FFT and Solvers for Exascale: FTFX and SpectralPACK

A first look


Front end

Hockney free-space convolution

Poisson's equation in free space

Partial differential equation (PDE)

Approach: Green's function

Solution characterization (.) Efficient through FFTs (frequency domain)

Technology + Results

FFT and Solvers for Exascale

Numerical linear algebra

- BLAS
- BLACS
- ScaLAPACK
- LAPACK

- Algorithm
- Low arithmetic intensity and variation of FFT use make library approach hard

- FFTW applications break down

- 3D problems themselves and then call the 1D FFT library

- Front-end guru

- Interface is powerful but hard to use, leading to performance loss

- Code optimization, etc., transparently used

- FFT code

- Graph algorithms

- Linear structures

- Algorithms: rules in domain-specific language

- Technology + Results

- Open source, available under open source

- SPIRAL 8.0: available under open source

- Open source SPMD

- Integrate applications on open source whole systems

- Commercial support on Spiral, Inc.

- Developed over 20 years

- Open SPIRAL with OpenACC back-end

- Parallelism and vectorization, etc., transparently used