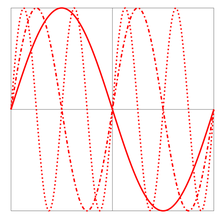
The Fast Fourier Transform in the Exascale Era

S. Aseeri[1], B.K. Muite[2], D. Takahashi[3]



Abstract

The Fast Fourier Transform (FFT) is an important component of many programs. On many emerging high performance computing architectures, the FFT may not work well on the full parallel computer. A good benchmark will lead to adoption of the best FFT software technology. Identification of alternative algorithms to the FFT along with comparisons of efficiency will lead to optimal use of high performance computers. A galvanized and involved benchmarking community is required to do this.

The Fast Fourier Transform

An accurate and low computational cost algorithm used for solving problems related to

- Wave propagation (such as seismic inversion)
- Diffusion (such as hydrocarbon reservoirs)
- Solid and fluid mechanics
- Electromagnetism
- Signal processing

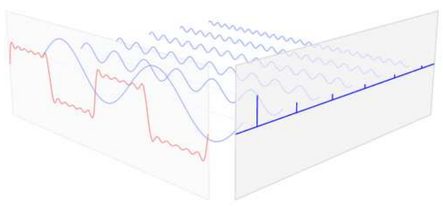
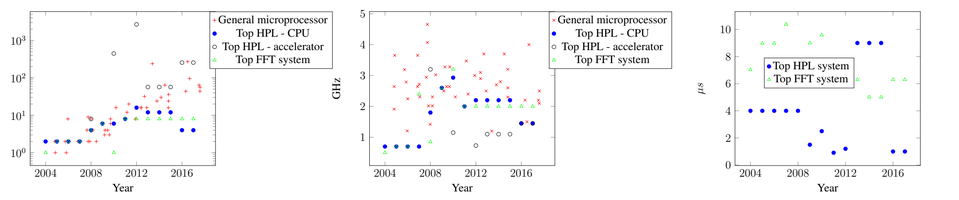


Figure 1: Demonstration of how the Fourier transform represents a signal in frequency space

Project Aim

- One cannot improve what one cannot measure
- Measure and report FFT performance to encourage improvements in efficiency and architectural adaptability
- Obtain consensus to enable widespead adoption of a long lived HPC benchmark suite, a component of which will enable performance prediction for FFT on high performance computers
- Create a benchmarking website resource that will serve as a guide for researchers and users of FFT libraries

Current Hardware Trends



(a) Cores on a chip (b) Processor clock speeds (c) Network zero byte latency

Figure 2: (a) and (b) compare general microprocessors (data from [3,4]) to the CPU and accelerator on the top system on the top 500 list and the top global FFT from HPC challenge. (c) shows latency of the top system on the top 500 list and the top global FFT from HPC challenge

- Fewer cores and lower clock speeds on the top systems than on the typical processor
- No improvement in network latency and low bisection bandwidth
- Heterogeneous hardware with accelerators such as GPUs (high bandwidth, even higher flop rate), NEC SX-Aurora TSUBASA (high bandwidth, balanced flop rate)
- $\rightarrow$ Need a flexible benchmark specification

Some Fast Fourier Transform Benchmarks

- **FFTW data comparison** (http://www.fftw.org/benchfft/)
  - Comparison of serial and multicore transforms.
  - Data presented as graphs on website
  - Not regularly updated
- **GearSHIFFT data comparison** (https://www.kcod.de/gearshifft/)
  - Recent benchmark
  - FFT benchmarking software that does multiple tests to estimate statistical variation
  - Currently focused on CPU (serial and multicore) and GPU
  - No distributed memory data
  - Nice web interface to explore data
  - Few submissions so far
- **HPC challenge** (http://icl.cs.utk.edu/hpcc)
  - Global one dimensional distributed memory FFT
  - Reference implementation uses FFTE
  - Data available from 2004, lower submission rate since 2012
- **NAS Parallel Benchmarks** (https://www.nas.nasa.gov/publications/npb.html)
  - Global three dimensional distributed memory FFT
  - Reference implementation uses Swartzrauber FFT
  - Little data available for download

Parallel Fast Fourier Transform Performance

- High performance computing challenge and high performance linpack

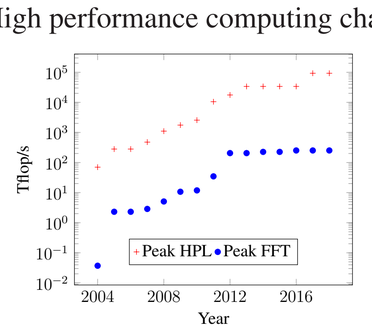


Figure 3: Evolution of best global 1D FFT performance from HPC challenge and best HPL performance from the Top 500 list

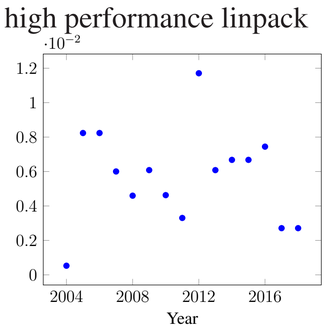


Figure 4: Evolution of ratio of best global 1D FFT performance from HPC challenge to best HPL performance from the Top 500 list

  - Ratio of relative performance of FFT to HPL is low
  - Global 1D FFT on K computer from 2011 still best of all HPC challenge submissions
- Benchmarking numerical solution of the Klein-Gordon equation

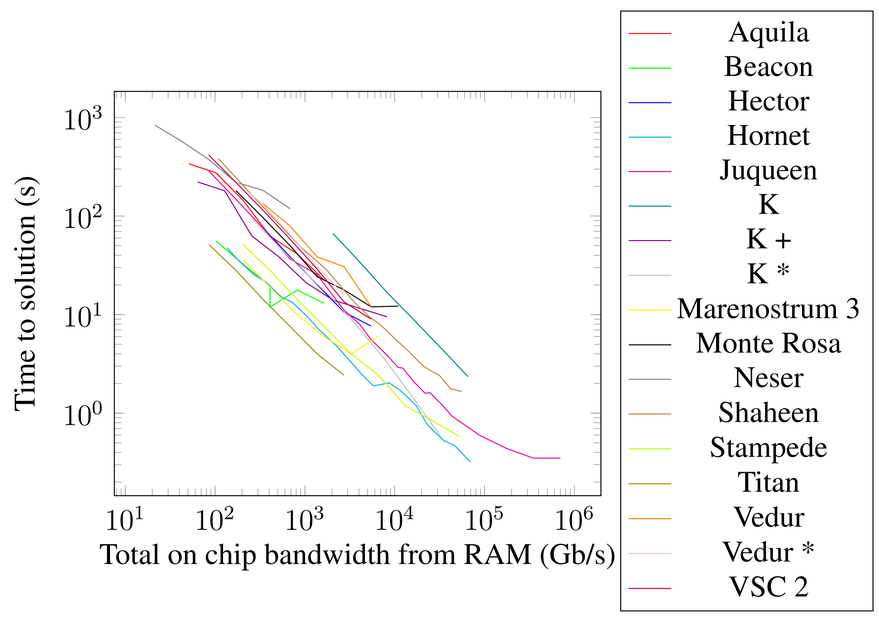


Figure 5: Time for solution of Klein Gordon equation using FFT on a 512^3 discretization [1]. Entries with a * used FFTE, all other entries used 2DECOMP&FFT. Entry with a + used 1 core per node.

  - Compare performance by best time to solution
  - Examine strong scaling $\rightarrow$ A larger computer on its own is not always helpful

Alternative Algorithms

- **Signal processing and linear solvers**: Sparse FFT, Non-uniform FFT
- **Linear solvers**: Fast Multipole Method, Multigrid, Fast Gauss Transform

Meetings To Date

- **Birds of Feathers**: SC 17 and ISC 18
- **Presentations**: SIAM PP 18 and IXPUG Middle East Conference 2018

Roadmap

By mid 2019, develop community and reach consensus on a benchmark or set of benchmarks to determine when to use the FFT and alternatives to the FFT on communication constrained parallel computer. After reaching consensus, foster community collectection and discussion of data.

- In person meetings at conferences related to high performance computing and domain specific areas that utilize parallel FFTs
- Collaboration with other benchmark writers to find a widely accepted and adopted benchmark suite for supercomputer co-design
- Online discussion and dissemination through web page (http://www.fft.report/), forum (https://www.forum.fft.report/) and mailing list (fft@lists.ut.ee)

References

1. Aseeri et al. "Solving the Klein-Gordon equation using Fourier spectral methods: A benchmark test for computer performance" April 2015
2. Bailey et al. "The NAS parallel benchmarks" March 1994
3. Frigo and Johnson "The design and implementation of FFTW" February 2005
4. Horowitz et al. "35 Years of microprocessor trend data" 2010
5. Luszczek et al. "Introduction to the HPC Challenge Benchmark Suite" March 2005
6. Rupp "42 Years of microprocessor trend data" February 2018
7. Steinbach and Werner "gearshifft – The FFT Benchmark Suite for Heterogeneous Platforms" May 2017

King Abdullah University of Science and Technology[1] University of Tartu[2] University of Tsukuba[3]