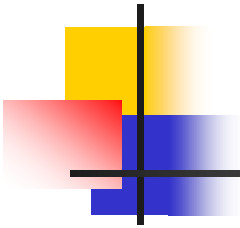
Discussion on FFTs in HPC

HECToR dCSE technical meeting

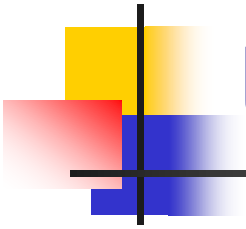
Manchester, 4-5 October 2011



dCSE Projects with FFT Work

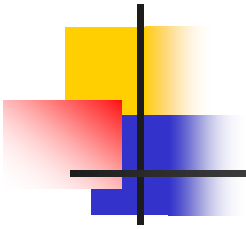
- Improving the performance of CP2K
- Scaling EBL turbulence application to thousands of cores
- Upgrading the FFTs in GS2
- Improving 3D FFT performance in GWW
- Incompact3D - DNS of turbulent flows
- ……

http://www.hector.ac.uk/cse/reports/



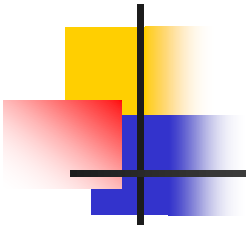
Use Library

- Do no rely on legacy code
- Use modern libraries: FFTW, ACML…
  - Or higher level libraries built on top of them
  - P3DFFT; 2DECOMP&FFT; CRAFFT…
- Libraries can be hard to use
  - FFTW advanced interface for C2C (fftw_plan_many_dft) has 13 parameters!
  - ACML expert driver (ZFFT3DY) has 17!
  - Again, use a higher level library
  - Or, at least, an abstraction layer in application



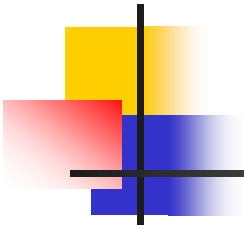
FFTW

- Version 3.3 release in July 2011
  - AVX support
  - Fortran 20003 interface
  - MPI interface reinstated



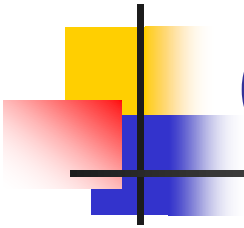
Parallel Strategy

- Threaded
  - FFTW, FFTE
- MPI
  - P3DFFT, 2DECOMP&FFT
- Hybrid
  - Way to go



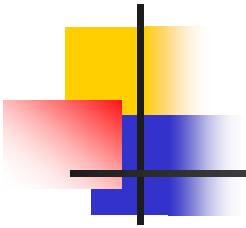
Data Distribution

- Parallel FFT algorithm
  - FFT on distributed data
  - 3D cube decomposition (like in DL_POLY)
- Transpose method - ALLTOALL(v)
  - 1D slab decomposition
  - 2D pencil decomposition
  - Special
    - CASTEP and Quantum ESPRESSO
    - Special storage requirement



Optimisations

- Point-to-point (preferably non-blocking)
- System V shared-memory communication
- Scatter/gather to emulate SHM
- One-sided communication
- Padded ALLTOALL optimisation
- Overlap of communications and computations
  - Using OpenMP threads
  - Non-blocking ALLTOALL(V)
- Flexible data layout (any i,j,k order; stride-1 layout)
- Hybrid MPI/threaded
- Combinations of some of the above

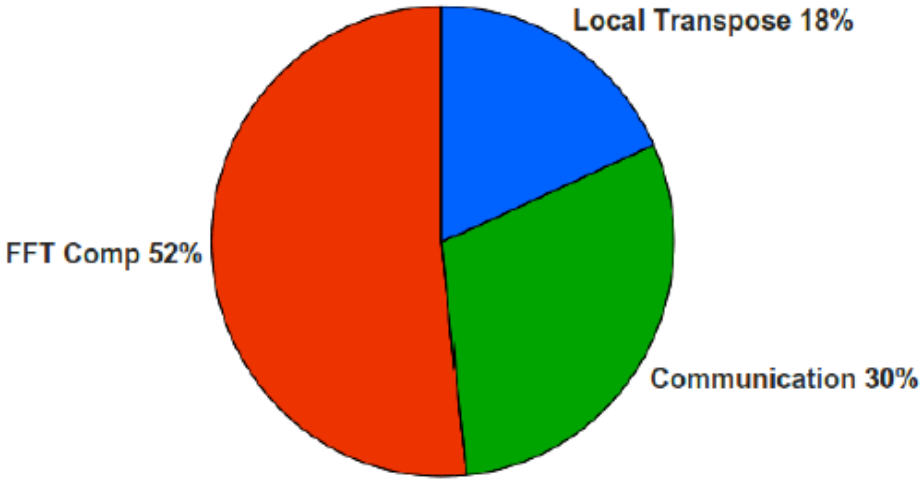


FFT on GPU

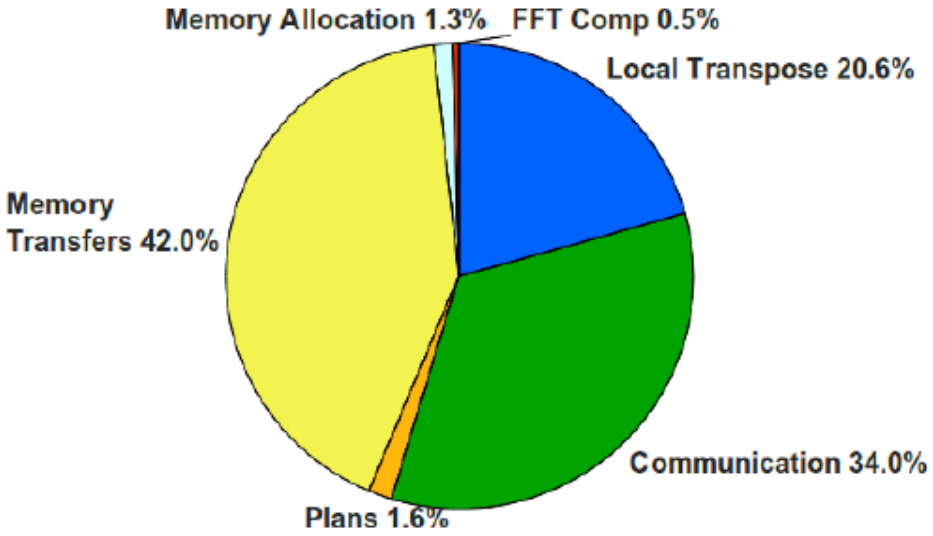
- CUFFT
  - API modelled after FFTW
  - Advanced interface
    - Only stride-1 data supported in version 3.2
    - Experimental C2C in version 4.0 that allows other memory pattern
- Distributed FFT on GPU cluster
  - Single digit speedup - not necessarily a good idea
  - digpufft (built for P3DFFT)
  - 2DECOMP&FFT has a CUFFT engine too.

FFT on GPU

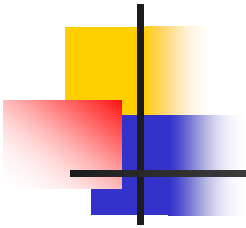
P3DFFT



DiGPUFFT



- FFT on 2048^3 mesh using 192 MPI ranks (McClanahan et al.)
- Completely different challenges on GPU
- Need to consider different algorithms
  - PKUFFT?



Discussions