

## Programming with Nonequispaced FFT

### Lab 3

### Parallel C Library Hands On

#### Exercise 1 (Installation of FFTW with MPI support):

Browse through the FFTW homepage

`http://www.fftw.org`

Then, download the newest FFTW package (version 3.3alpha1) and build the library in your home directory on JUMP, i.e.,

1. `ssh userid@jump.fz-juelich.de`
2. `bash`
3. `mkdir src; cd src`
4. `wget http://www.fftw.org/fftw-3.3alpha1.tar.gz`
5. `gtar xzvf fftw-3.3alpha1.tar.gz`
6. `cd fftw-3.3alpha1`
7. `export CC="xlc" CFLAGS="-O3 -q64 -qhot" AR="ar -X64"`
8. `export F77="xlf -q64" FFLAGS="-O3 -qhot -qstrict" MPICC="mpicc -q64"`
9. `./configure --enable-mpi --prefix=$HOME/FFTW3_3_MPI`
10. `gmake all install`

Since there is an older FFTW installed on JUMP, rename the serial FFTW library, i.e.

1. `cd ~/FFTW3_3_MPI/lib`
2. `ln -s libfftw3.a libfftw3_3_alpha1.a`

Now linking to `libfftw3_3_alpha1.a` assures that we use the newest FFTW library.

### Exercise 2 (Installation and testing of PFFT):

Download the PFFT package from <http://www.tu-chemnitz.de/~mpip/software> and build the library in your home directory on JUMP, i.e.,

1. `cd ~/src`
2. `wget http://www.tu-chemnitz.de/~mpip/software/pfft-1.0alpha1.tar.gz`
3. `gtar xzvf pfft-1.0alpha1.tar.gz`
4. `cd pfft-1.0alpha1`
5. `ln -s makeinc.jump makeinc`
6. `gmake all install`

Lookup and open the source file `simple_test.c`. Skim through the main routine. Try to understand what it does. Then, run the actual executable `simple_test.x` found in `~/PFFT/examples` with 4 processors, i.e., `llrun -p4 ~/PFFT/examples/simple_test.x`.

Lookup and open the source file `simple_test.c` again. Add timing measurements of the subroutines `pfft_execute(fftwplan_forw)`, `pfft_execute(fftwplan_back)`. Repeat the build process and run `simple_test.x` again.

Hint: Use the function `MPI_Wtime()` to get times.

### Exercise 3 (Installation and testing of PNFFT):

Download the PNFFT package from <http://www.tu-chemnitz.de/~mpip/software> and build the library in your home directory on JUMP, i.e.,

1. `cd ~/src`
2. `wget http://www.tu-chemnitz.de/~mpip/software/pnfft-1.0alpha1.tar.gz`
3. `gtar xzvf pnfft-1.0alpha1.tar.gz`
4. `cd pnfft-1.0alpha1`
5. `ln -s makeinc.jump makeinc`
6. `gmake all install`

Lookup and open the source file `simple_test.c`. Skim through the main routine. Try to understand what it does. Then, run the actual executable `simple_test.x` found in `~/PNFFT/examples` with 4 processors, i.e., `llrun -p4 ~/PNFFT/examples/simple_test.x`.

Lookup and open the source file `simple_test.c` again. Add timing measurements of the subroutines `pnfft_trafo(&nfft)`, `pnfft_adj(&nfft)`. Repeat the build process and run `simple_test.x` again.

Hint: Use the function `MPI_Wtime()` to get times.