PARALLEL CODE: FOURIER TRANSFORM ACCELERATION

- Spend 90% of time doing FFTs!
- Therefore optimise carefully for each machine
- Use vendor supplied FFTs
- Use shared memory whenever possible
- Avoid hot spots in communication (scheduling)

**e.g. FORWARD TRANSFORM**

DO my_z = my_z%min , my_z%max

**local fft over x:**

$(x, y, z) \rightarrow (kx \_ y, z)$

real $\rightarrow$ complex

real_field$(x, y, my_z) \rightarrow$ complex_field$(nx/2+1, ny, my_z)$

**local transpose (including x−dealias):**

$(kx, y, z) \rightarrow (y, kx, z)$

complex_field$(nx/2+1, ny, my_z) \rightarrow$ complex_field$(ny, nx\_dealias/2+1, my_z)$

**local fft over y:**

$(y, kx, z) \rightarrow (ky, kx, z)$

(complex $\rightarrow$ complex)

complex_field$(ny, nx\_dealias/2+1, my_z) \rightarrow$ complex_field$(ny, nx\_dealias/2+1, my_z)$

**y−dealias and re−pack:**

complex_field$(my_z, ny\_dealias, nx\_dealias/2+1)$

END DO

**Distributed transpose:** I have all kxy for certain z. Send right kxy's to the right processor

DO pp = 1, N_p

$p = \text{schedule}(\text{mod(mype+p, N_p)})$ randomise send scheduling

SEND : Processor p gets sent

complex_field$(all\_my\_nz, p*nxy\_dealias/N_p: (p+1)*nxy\_dealias/N_p - 1)$

END DO

BARRIER

DO p = 1, N_p

RECEIVE : Processor p receives

complex_field$(1+nz*p/N_p: nz*(p+1)/N_p, all\_the\_kxy)$

END DO